Latent Probabilistic Topic Discovery for Text Documents Incorporating Segment Structure and Word Order

JAMEEL, Mohammad Shoaib

A Thesis Submitted in Partial Fulfilment

of the Requirements for the Degree of

Doctor of Philosophy

in

Systems Engineering and Engineering Management

The Chinese University of Hong Kong

July 2014

بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ

Thesis/Assessment Committee

Professor CHENG, Chun Hung (Chair)

Professor LAM, Wai (Thesis Supervisor)

Professor MENG, Mei Ling Helen (Committee Member)

Professor Michael Chau (External Examiner)

Preface

This dissertation is submitted for the degree of Doctor of Philosophy at The Chinese University of Hong Kong. The research described herein was conducted under the supervision of Prof. LAM, Wai, between August 2009 and May 2014. This work is to the best of my knowledge original, except where acknowledgment and reference is made to previous work. Neither this, nor any substantially similar dissertation has been or is being submitted for any degree, diploma or other qualification at any other university. Part of this work are published (and some under review) in the following publications:

1. Lidong Bing, Wai Lam, **Shoaib Jameel**, and Chunliang Lu. "Website Community Mining from Query Logs with Two-Phase Clustering." In *Computational Linguistics and Intelligent Text Processing* (CICLing), pp. 201–212. Springer Berlin Heidelberg, 2014.

2. **Shoaib Jameel**, and Wai Lam. "A Nonparametric N-Gram Topic Model with Interpretable Latent Topics." In *Proceedings of the Ninth Asia Information Retrieval Societies Conference* (AIRS), pp. 74–85. Springer Berlin Heidelberg, 2013.

3. **Shoaib Jameel**, and Wai Lam. "An unsupervised topic segmentation model incorporating word order." In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval* (ACM-SIGIR), pp. 203–212. ACM, 2013.

4. **Shoaib Jameel**, and Wai Lam. "An N-gram topic model for time-stamped documents." In *Proceedings of the 35th European Conference on Information Retrieval* (ECIR), pp. 292–304. Springer Berlin Heidelberg, 2013.

5. **Shoaib Jameel**, Xiaojun Qian, and Wai Lam. "N-gram Fragment Sequence Based Unsupervised Domain-Specific Document Readability". In *Proceedings of the 24th International Conference on Computational Linguistics* (COLING), pp. ACL, 1309–1326, 2012.

6. **Shoaib Jameel**, Wai Lam, Xiaojun Qian, and Ching-man Au Yeung. "An unsupervised technical difficulty ranking model based on conceptual terrain in the latent space." In *Proceedings of the 12th ACM/IEEE-CS Joint conference on Digital Libraries* (ACM-JCDL), pp. 351–352. ACM, 2012.

7. **Shoaib Jameel**, and Xiaojun Qian. "An Unsupervised Technical Readability Ranking Model by Building a Conceptual Terrain in LSI." In *Eighth International Conference on Semantics, Knowledge and Grids* (SKG), pp. 39–46. IEEE, 2012.

8. **Shoaib Jameel**, Wai Lam, and Xiaojun Qian. "Ranking Text Documents Based on Conceptual Difficulty using Term Embedding and Sequential Discourse Cohesion." In *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01* (WI-IAT), pp. 145–152. IEEE Computer Society, 2012.

9. **Shoaib Jameel**, Wai Lam, Ching-man Au Yeung, and Sheaujiun Chyan. "An unsupervised ranking method based on a technical difficulty terrain." In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management* (ACM-CIKM), pp. 1989–1992. ACM, 2011.

10. **Shoaib Jameel**, Wai Lam, and Lidong Bing. "Nonparametric Topic Models with Word Order for Text Documents." *Journal paper - Under review.*

11. **Shoaib Jameel**, Wai Lam, and Lidong Bing. "Supervised Topic Models with Word Order Structure for Document Classification and Retrieval Learning."

Conference paper - Under review.

12. **Shoaib Jameel**, and Wai Lam. "N-gram Fragment Based Domain-Specific Readability Ranking Model for Text Documents." *Journal paper - Under review.*

Environment Friendly Thesis

I have found a common limitation in most of the theses that I have read so far. The authors of those theses do not make their document environment friendly. What I mean by this is, one is forced to print the thesis for smooth reading. This is exemplified by the following scenario:

Imagine that you are reading a long thesis written by a person named Dr. Doolittle. Since you don't have a printer with you or you want to save some papers by refraining yourself from printing a long thesis, you try to read it on your computer. However, while reading the content and upon reaching some citation in the thesis, you are forced to check what research paper it is actually referring to. Then you scroll down the thesis to look for the citation. You view that citation, and again try to locate the specific location in the document where you were reading it previously. Repeatedly doing this results in a loss of interest, and thus forces you to make your task simpler. Therefore, a better option then is to print the thesis. Some clever ones will only print the references section and read the text from the computer. But still something is printed and paper is lost.

I have made this thesis as environment friendly as I could. It means that I have worked extra hard to discourage people from printing it, considering that it is a long thesis. I have included special effects in this electronic version of the thesis, where hovering the mouse pointer over any citation pops-up the full reference .

One can click on that pop-up text window, and place it anywhere one likes in the text window. Remember to click the pop-up text window once and then release the mouse pointer. This will make the text window stick to the mouse pointer. Same is done for all abbreviations in this thesis. This will make the process of reading this thesis pleasant and smooth. Most importantly, paper is saved. One can argue about the heat generated by the computer monitor while reading this thesis and its effect on the environment. There are always some trade-offs in everything that one does.

JAMEEL, Mohammad Shoaib

Dedication

I would like to dedicate this small piece of work to my parents, and to all those who love and care for me. Without such wonderful people around me, it would not have been possible for me to learn many things not only in Science, but even in general aspects of life in the course of these five years and all throughout my life.

I would also dedicate my "hard-work" to all those innocent poverty-stricken people all across the world who sleep with empty stomach at night hoping that next day an agent from beloved God would descend down to Earth and bless them with food so that they never have to go back to sleep without it. I hope my hard-work could help them in the future, and I can one day become that agent from God.

Whatever you bestow in charity must go to parents and to kinsfolk, to the orphans and to the destitute and to the traveler in need. - The Glorious Quran, The Cow 2:215

Let him never turn away (a stranger) from his house, that is the rule. Therefore a man should by all means acquire much food, for (good) people say (to the stranger): There is food ready for him. - Taittiriyaka Upanishad, 3rd Valli, 10th Anuvaka:I

For I was hungry, and you gave me something to eat; I was thirsty, and you gave me something to drink; I was a stranger, and you took me in; I was sick and you took care of me, I was in prison and you visited me...I tell you whenever you do these things for the least important of these followers of mine, you did it for me. - Jesus: Matthew 25:35-40

The generous man is blessed. For he gives of his bread to the poor. - Old Testament, Proverbs 22:9

Acknowledgements

I would like to express my heartfelt thanks to my parents, other members of my family, and my supervisor Prof. LAM, Wai, whose encouragement and support has made me do this work possible. Moreover, I would like to emphasize the human values like simplicity and elegance that I found in Prof. LAM, Wai that has given me the inspiration to work in the Text Mining Group of the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong. Hence, I would like to add a note that it has not only been scientifically beneficial for me, but I have also benefited as a person.

I can never forget my teachers who have taught me right from the very beginning. Miss. Prema Subramaniam taught me to read and write which is useful to me until this day. Thanks to my Mathematics teacher, Miss. Oli Ganguly for teaching me basic Mathematics which I am still applying today and publishing papers out of it. Thanks to my undergraduate research mentors Prof. Tejbanta Singh Chingtham, Professor at the Sikkim Manipal Institute of Technology, Sikkim, India, Dr. Marimuthu Muruganant (Cantab.), Mr. Fredi B. Zarolia from Tata Steel Limited, Jamshedpur, India, whose love and support has helped me reach where I am today. Thanks to all my other teachers both in my school and undergraduate university. Special mention of Mr. Andrei Raevsky from New Smyrna Beach, Florida for helping me proof-read my research and personal statements for applications to different universities for doctoral study. Special thanks to my thesis committee which comprises of Prof. Helen Meng and Prof. C.H. Cheng whose inputs during the course of my doctoral study proved useful.

Due acknowledgment goes to Xiaojun Qian, Doctoral student at the Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong without whose valuable discussions and help in the early stages of my thesis work, this thesis would not have been been written. Thanks to Dr. Wei Gao, Scientist, Qatar Computing Research Institute for his friendship and discussions. In addition, thanks to Ching-man Au Yeung for collaborating in some of my early works.

I would also thank the present and past members of my Text Mining Group, in particular, Chen Bo and Lidong Bing who have played an instrumental role during the early stages of my research life. Also, thanks to Chunliang Lu for helping me solve some programming related problems. Members from the other research groups in my department indeed need to be thanked, in particular, Li Binyang for proof-reading some of my research articles. Merak Cheung with whom I used to have research discussions. Haiqin Yang from the Department of Computer Science and Engineering, The Chinese University of Hong Kong for helping me during the initial research study. Thanks to Priyanka Garg from the Department of Computer Science and Engineering, The Chinese University of Hong Kong, with whom I used to have discussions related to some raw research ideas, and how to pursue such ideas for future publications.

Writing acknowledgment without mentioning about the people whom you have met during conferences is unthinkable. I am thankful to Dmitry Ignatov from the Higher Secondary of Economics, Moscow, Russia for his help during my stay in the Russian Federation, and also explaining to me the basic concepts of Formal Concept Analysis (FCA). Other folks from Russia need to be specially thanked in this thesis because of their wonderful support during my trip to Russia. Many thanks to Alexandra Kaminskaya, Yandex, Moscow whose help before my conference trip to the Russian Federation proved very fruitful. I would like to specially thank Karina Gaynutdinova student at the Higher Secondary of Economics, Moscow for her help

during my stay in Moscow. Without all these wonderful Russian people, conference trip to Russia would have been a nightmare. In addition, thanks to people from Yandex, Moscow, in particular, Eugene Kharitonov, Researcher at Yandex for having some technical discussions about Yandex and its business model. Other people with whom I had productive technical discussions, and who are worth mentioning are Prof. Stephen Robertson, currently a visiting Professor at University College London, Department of Computer Science, with whom I had discussions regarding the future of web search and how the query log data could help simplify web search algorithms. Van Dang from the University of Massachusetts, Amherst (currently at Google) with whom I had discussions about the learning-to-rank models. Prof. Mark Sanderson from Royal Melbourne Institute of Technology, Australia, with whom I had some insightful research discussions related to text readability and its application to web search. This discussion happened even before I began my doctoral study. Henry Field from the University of Massachusetts, Amherst (currently Assistant Professor of Computer Science at Endicott College in Beverly, MA) with whom I had discussions about writing a research paper without using lot of Mathematics, and how to get it accepted in conferences such as SIGIR. Elif Aktolga from the University of Massachusetts, Amherst (currently at Apple, Inc.) for teaching me how to find out whether a steak is properly cooked or not. Dell Zhang from the Birkbeck College, University of London, who gave some useful comments about my research. Antonio Gulli, Research Scientist in Web Information Retrieval at Microsoft, for asking me (tough) questions consistently during my talks in a couple of conferences. Co-incidentally, I used to find him in all my talks, if we are at the same conference. I would also thank others whom I met for a short while.

Many thanks to the Department of Systems Engineering and Engineering Management for funding my doctorate study. Also to the Research Grant Council of the Hong Kong Special Administrative Region, China, Microsoft-CUHK Joint Laboratory for Human-Centric Computing and Interface Technologies, Yandex and Association for Computing Machinery for funding my conference trips.

Abstract

Probabilistic topic models are a class of statistical techniques that bring out the latent representation in the data. These models are able to find new data correlations in a low-dimensional topic space. Unigram based probabilistic parametric topic models assume that the order of words in a document is not important. As a result, these models lose important structural information which is inherent in the document. This results in the generation of less interpretable topics with ambiguous words. In addition, such parametric topic models assume a pre-defined parameter space i.e. they cannot grow with the data complexity. This is an important limitation to address, as in reality a user does not know the appropriate parameter space of the data a priori. Many topic models also do not consider the temporal aspect of the data with word order. So they fail to capture how topics evolve over time in the data, as large data is usually collected over time, and topics rise and fall as time passes by. In addition, another limitation of unsupervised topic models for document classification is that the topic models do not consider useful side information with word order, for instance, class labels of documents along with the word order structure in text documents that could interact with the class label information for solving the document classification task. Likewise, supervised topic models with word order structure have not been explored in document retrieval learning where relevance assessments given by human annotators can be used as a side information in the topic model itself.

This thesis presents novel structured topic models for text data which address the above shortcomings. The models proposed in this thesis maintain the word order structure of a text document. In addition, document structure such as paragraphs and sentences can be maintained. The prime motivation is that word order helps capture the semantic nature of text better than the unigram models, as it models the way humans write and read documents. In addition, it can capture both long and short range text co-occurrences. In doing so, the proposed models in this thesis

obtain state-of-the-art results in several text mining tasks such as document retrieval learning and document classification. Qualitatively, the models generate better topical phrasal words than the previously proposed models. This thesis presents some of the groundbreaking work in structured topic discovery which strengthens the claim that capturing the document's structure is of utmost importance rather than make strong bag-of-words assumption.

Abstract

概率主题模型（probabilistic topic models）是一类统计方法，其可以生成数据的潜在表示（latent representation）。这类模型可以在低维空间中挖掘数的新关系。基于一元模型（unigram）的带参概率主题模型（ probabilistic parametric topic models）假设文档中词的顺序是不重要。该假设导致此类模型丢失了文档中固有的结构信息。因此，此类模型会生成难以解释的主题。此外，此类带参主题模型假设一个预定义的参数空间，此参数空间不随数据复杂度的变化而改变。然而现实中人们无法预先知道某一数据集合的合适参数空间 很多已有的主题模型也不考虑数据的时间属性（temporal aspect）。大数据集合随着时间被逐步收集，主题随着时间消长，而上述模型不能描述主题随着时间变化的演进。此外，非监督主题模型（unsupervised topic models）应用于文档分类（document classification）的另一个局限是，此类模型不考虑有用的词序（word order）信息。而文档的类别标签和文档内词序结构的信息交互作用，可以为更好的解决分类问题提供支持。考虑词序的监督主题模型（supervised topic models）尚未被应用到文档检索学习（document retrieval learning）中，而此处可将标注人员给定的相关性判断（relevance assessments）应用于主题模型本身。

本论文描述了一组富有新意的结构化主题模型（structured topic models），可应用这些模型来解决以上文本数据挖掘的局限性。本文提出的模型保持了文本的词序结构信息。同时，其他文档结构如段落、句子信息也被保持。提出这组模型的核心思想是词序信息可以比一元模型更好的表述文档的语义本质（semantic nature），因为词序可以对人们读写文本的方式进行建模。此外，词序还可以表述长、短跨度的文本共现关系。由于这些优点，本论文提出的模型取得了在若干文本挖掘任务下的当前最优结果，如文本检索学习和文本分类。定性分析上，相较于以前的模型，本文的模型生成了更好的主题短语词（topical phrasal words）。本文描述了结构化主题发现（structured topic discovery）的开创性工作，印证了表述文档结构，而非做强词袋假设（strong bag-of-words assumption），是极其重要的这一结论。

Contents

Preface iv

Dedication viii

Acknowledgments ix

Abstract xii

Abstract in Chinese xiv

Contents xv

List of Tables xix

List of Figures xxi

List of Algorithms xxix

List of Abbreviations xxx

1 Introduction 1
- 1.1 Motivation . . . 2
- 1.2 Applications of Topic Models . . . 8
- 1.3 Contributions . . . 9
- 1.4 Outline . . . 11

2 Literature Survey 13
- 2.1 Unsupervised Parametric Topic Modeling with Exchangeability Assumption . . . 16
- 2.2 Unsupervised Parametric Topic Modeling with Word Order . . . 19
- 2.3 Unsupervised Nonparametric Topic Modeling with Exchangeability . . . 23
- 2.4 Unsupervised Nonparametric Topic Modeling with Word Order . . . 24

2.5 Unsupervised Parametric Topic Models for Temporal Data 25
2.6 Supervised Parametric and Nonparametric Topic Models 27
2.7 Supervised and Unsupervised Readability Prediction Models 30

3 Background **37**
3.1 Cluster Analysis 38
3.2 Principal Component Analysis (PCA) 41
3.2.1 Singular Value Decomposition (SVD) 42
3.3 Latent Class Analysis (LCA) 43
3.4 Latent Semantic Analysis (LSA) 44
3.4.1 Limitations of the Latent Semantic Analysis (`LSA`) Model . . . 47
3.5 Probabilistic Latent Semantic Analysis (pLSA) 47
3.5.1 Probabilistic Latent Semantic Analysis (`pLSA`) as a Matrix Factorization Model 49
3.6 Dirichlet Distribution 51
3.7 Probabilistic Unsupervised Topic Modeling 54
3.7.1 Latent Dirichlet Allocation Model 54
3.7.2 Latent Dirichlet Allocation (`LDA`) as a Matrix Factorization Scheme 60
3.8 Topic Models with Word Order 61
3.8.1 Bigram Topic Model (`BTM`) 61
3.8.2 LDA-Collocation Model (Latent Dirichlet Allocation-Collocation (`LDACOL`)) 63
3.8.3 Topical N-gram Model (`TNG`) 67
3.9 Topic Segmentation Models 71
3.10 Bayesian Nonparametrics in Topic Modeling 77
3.10.1 Dirichlet Processes (DP) 77
3.10.2 Stick Breaking Construction 82
3.10.3 The Dirichlet Process Mixture Model (DPMM) 84
3.10.4 Chinese Restaurant Process (CRP) 86
3.10.5 Hierarchical Dirichlet Processes 87
3.11 Supervised Topic Models 92
3.12 MedLDA Model 92
3.12.1 Posterior Inference using Gibbs Sampling 100
3.13 Learning-to-Rank 102
3.13.1 Features 103

4 Topic Segmentation Model for Text Documents **104**
4.1 The Case for Topic Segmentation with Word Order 105
4.2 N-gram Topic Segmentation Model Description 108
4.3 Posterior Inference 113
4.4 Experiments and Results 118
4.4.1 Correlation Graph 119
4.4.2 Topic Segmentation Experiment 120
4.4.3 Document Classification Experiment 128

4.4.4 Document Likelihood Experiment . . . 132
4.5 Closing Remarks . . . 134

5 Modeling Temporal Dynamics in Text Documents 135
5.1 The Case for Capturing N-grams over Time . . . 136
5.2 Our N-gram Temporal Topic Model . . . 138
5.2.1 Inference and Parameter Estimation . . . 140
5.3 Experiments and Results . . . 142
5.3.1 Data Sets and Comparative Method . . . 142
5.3.2 Experimental Results . . . 145
5.4 Closing Remarks . . . 154

6 Bayesian Nonparametric Topic Models for Text Data 155
6.1 The Case for Bayesian Nonparametric Topic Models With Order . . . 156
6.2 Nonparametric N-gram Collocation Model . . . 158
6.3 Posterior Inference . . . 160
6.3.1 The First Condition: . . . 161
6.3.2 The Second Condition: . . . 162
6.3.3 Sampling the Concatenation Indicator Variables: . . . 163
6.4 Nonparametric N-gram Topic Models (NNTM) . . . 164
6.4.1 Model Description of NNTM-1 . . . 165
6.4.2 Model Description of NNTM-2 . . . 175
6.5 Experiments and Results . . . 181
6.5.1 Document Modeling Experiments . . . 181
6.5.2 Running Time Comparison . . . 190
6.5.3 Qualitative Results . . . 192
6.5.4 Document Classification Experiments . . . 196
6.6 Closing Remarks . . . 202

7 Supervised Probabilistic Topic Models 203
7.1 The Case for the Supervised Topic Models With Word Order . . . 204
7.2 Our Classification Model . . . 205
7.2.1 Model Description . . . 205
7.2.2 Posterior Inference . . . 209
7.3 Document Classification Experiments . . . 211
7.3.1 Experimental Setup . . . 211
7.3.2 Quantitative Results . . . 214
7.3.3 Qualitative Results . . . 217
7.4 Closing Remarks . . . 218

8 Document Retrieval Learning Models 219
8.1 The Case for the Supervised Document Retrieval Learning Topic Model 220
8.2 Model Description . . . 221
8.2.1 Posterior Inference . . . 230
8.2.2 Ranking Unseen Documents . . . 232

8.3 Retrieval Learning Experiments . . . 233
8.3.1 Experimental Setup . . . 233
8.3.2 Quantitative Results . . . 236
8.3.3 Qualitative Analysis . . . 238
8.4 Closing Remarks . . . 239

9 Readability Prediction and Ranking 240
9.1 The Case for the Terrain Models in LSI . . . 241
9.2 The Terrain Model in the Concept Space . . . 245
9.2.1 Sequential Term Transition Model (STTM) . . . 245
9.2.2 Document Conceptual Difficulty Score . . . 252
9.2.3 Experiments and Results . . . 253
9.2.4 Results Discussion . . . 257
9.3 Closing Remarks . . . 260

10 Conclusions and Future Directions 261
10.1 Summary of the Methods . . . 262
10.2 Shortcomings of the Models . . . 263
10.3 Suggestions for Future Research . . . 264
10.4 Personal Research Experience . . . 266

A N-gram Topic Segmentation Model - Full Gibbs Sampling Derivation 268

B N-gram Topics Over Time Model - Full Gibbs Sampling Derivation 273

C Bigram Topic Model - Full Gibbs Sampling Derivation 277

D LDA-Collocation Model - Full Gibbs Sampling Derivation 280

E Proof of Bayes' Theorem when Cast into an Optimization Problem 283

F MedLDA Model - Full Collapsed Gibbs Sampling Derivation 286

G Proof of Bayes' Theorem when Cast into an Optimization Problem for Our Bigram Supervised Topic Model 290

H Bigram Supervised Topic Model - Full Gibbs Sampling Derivation 293

Bibliography 296

List of Tables

1.1 Example of n-grams. We only depict unigrams, bigrams and trigrams. 6

4.1 Classification results for N-gram Topic Segmentation (NTSeg) - Computer Dataset . . . 127
4.2 Classification results for NTSeg - Science Dataset . . . 127
4.3 Classification results for NTSeg - Politics Dataset . . . 128
4.4 Classification results for NTSeg - Sports Dataset . . . 128

5.1 Decade prediction results of our N-gram Topics Over Time (NTOT). . 151

6.1 Datasets used in evaluation of parametric and nonparametric topic models. . . . 182
6.2 Perplexity results for parametric and nonparametric topic models. . . 185
6.3 Topics obtained from the tuning process in the parametric topic models. 186
6.4 Qualitative results for the topic "technology" from AQUAINT-1 for various nonparametric topic models. . . . 194
6.5 Qualitative results for the topic "war" from AQUAINT-1 for various nonparametric topic models. . . . 194
6.6 Qualitative results for the topic "neural networks" from NIPS for various nonparametric topic models . . . 194
6.7 Qualitative results for the topic "speech technology" from NIPS for various nonparametric topic models . . . 194
6.8 Qualitative results for the topic "cells" from OHSUMED for various nonparametric topic models . . . 195
6.9 Qualitative results for the topic "liver" from OHSUMED for various nonparametric topic models . . . 195
6.10 Qualitative results for the topic "finance" from Reuters for various nonparametric topic models . . . 195
6.11 Qualitative results for the topic "oil" from Reuters for various nonparametric topic models. . . . 195
6.12 Document classification dataset used in nonparametric topic model experiments. . . . 197
6.13 Classification results on the Computer Dataset using parametric and nonparametric topic models. . . . 198
6.14 Classification results on the Science Dataset using parametric and nonparametric topic models. . . . 198

6.15 Classification results on the Politics Dataset using parametric and nonparametric topic models. 199
6.16 Classification results on the Sports Dataset using parametric and nonparametric topic models. 199
6.17 Classification results on the OHSUMED Dataset using parametric and nonparametric topic models. 199

7.1 Classification performance of our supervised topic model. 214
7.2 The effect of the number of topics on document classification measured by F-measure. 215
7.3 Top five probable words from a topic from *comp.graphics* class of 20 Newsgroups dataset. 217

8.1 Features used in our discriminant function in our document retrieval learning model. 229
8.2 The performance of our model in comparison to other learning-to-rank models. 237
8.3 Results obtained from our models when the number of topics is varied. 238
8.4 Top five probable words from a topic from AQUAINT-1 collection. . 239

9.1 Readability annotation guidelines to the human annotators. 256
9.2 Ranking performance of the traditional Information Retrieval (IR) models in solving the readability problem. 257
9.3 Ranking performance of different readability models in the Psychology domain. 258
9.4 Query-wise performance of different readability models in the Psychology domain. 258

List of Figures

1.1 Input, process and output of a bag-of-words unsupervised topic model. 3
1.2 An example of two topics formed by the LDA model. 4
1.3 Word ambiguity as a comical representation. 7
1.4 Figure depicting the applications of probabilistic topic models. 8

3.1 An illustration of cluster analysis. 39
3.2 Mixture model example. 40
3.3 Hierarchical and partitional clustering. 40
3.4 Principal components 41
3.5 An example of Latent Class Analysis 44
3.6 pLSA graphical model. 48
3.7 Simulating the Dirichlet with different α 52
3.8 Simulating the Dirichlet with different α 52
3.9 Simulating the Dirichlet with different α 53
3.10 Latent Dirichlet Allocation model graphical model 54
3.11 Explanation of the LDA graphical model. 56
3.12 Diagrammatic representation of a generative process 57
3.13 Diagrammatic representation of an inference process 57
3.14 Topic visualization using the LDA model 59
3.15 Bigram Topic Model 62
3.16 Graphical model of the LDACOL model. 64
3.17 Graphical model of the Topical N-gram (TNG) model 68
3.18 Modified TNG model. 69
3.19 The graphical model of the Latent Dirichlet Allocation Segmentation (LDSEG) model in a plate diagram. 73
3.20 The graphical model of the LDSEG model in a plate diagram highlighting the portion that generates the super-topics. 74
3.21 The graphical model of the LDSEG model in a plate diagram highlighting the portion that generates the word-topics. 75
3.22 An illustration how the LDSEG model segments a document into paragraphs. 76
3.23 An example showing partition of data. 78
3.24 Density plots for the Polyà Urn model for different α values. 79
3.25 Density plots for the Polyà Urn model for different α values. 80
3.26 Density plots for the Polyà Urn model for different α values. 81
3.27 Stick breaking construction representation 82

3.28 Stick breaking construction representation . . . 83
3.29 Stick breaking construction representation . . . 83
3.30 Dirichlet Process Mixture Model (DPMM) graphical model. . . . 85
3.31 The Hierarchical Dirichlet Processes (HDP) model in different metaphors. 88
3.32 Graphical model of the Maximum-Margin Entropy Discrimination Latent Dirichlet Allocation (MedLDA) model. . . . 93
3.33 Graphical model of the MedLDA model when expanded to show the words in the document. . . . 94
3.34 High-level illustration of the learning-to-rank framework . . . 102

4.1 Our NTSeg model in plate notation . . . 106
4.2 Our NTSeg model depicting the portion which generates segments . . 107
4.3 Our NTSeg model depicting the portion that generates n-gram words 107
4.4 N-gram word generation . . . 108
4.5 Depiction of longer phrase generation . . . 110
4.6 Word-topic and segment-topic illustration . . . 110
4.7 Correlation graph of NTSeg . . . 120
4.8 Correlation graph of NTSeg . . . 121
4.9 Correlation graph of Correlated Topic Model (CTM) . . . 122
4.10 Topic segmentation results . . . 123
4.11 Topic segmentation results . . . 124
4.12 Topic segmentation results . . . 125
4.13 Topic segmentation results . . . 126
4.14 Document modeling results of NTSeg . . . 129
4.15 Document modeling results of NTSeg . . . 130

5.1 An illustration of topic change over time. . . . 137
5.2 NTOT model . . . 138
5.3 Histograms and topical words from NTOT model for the topic "Mexican War" . . . 142
5.4 Histograms and topical words from Topics Over Time (TOT) model for the topic "Mexican War" . . . 143
5.5 Histograms and topical words from NTOT model for the topic "Panama Canal" . . . 143
5.6 Histograms and topical words from NTOT model for the topic "Panama Canal" . . . 144
5.7 "Recurrent NNs" topic over time depiction via our model. . . . 146
5.8 "Recurrent NNs" topic over time depiction via the TOT model. . . . 146
5.9 Topical words as they change over time in our model. . . . 148
5.10 Topical words as they change over time in the TOT model. . . . 149
5.11 Alternative depiction of the NTOT model. . . . 150
5.12 Co-occurrence with "classification" topic over time in TOT model. . . 152
5.13 Co-occurrence with "classification" topic over time in NTOT model. . . 153

6.1 Nonparametric N-gram HDP model. . . . 159
6.2 Graphical model of the NNTM-1 model. . . . 165
6.3 Chinese Restaurant Franchise (CRF) with Buddy Customers . . . 168
6.4 Our proposed NNTM-2 in Chinese Restaurant Franchise representation 176
6.5 The effect of the topic Dirichlet parameter in nonparametric topic models. . . . 187
6.6 Number of topics detected by nonparametric topic models in NIPS collection. . . . 188

6.7 Number of topics detected by nonparametric topic models in OHSUMED. 188
6.8 Number of topics detected by nonparametric topic models in Reuters. 188
6.9 Number of topics detected by nonparametric topic models in AQUAINT-1. 189
6.10 Training and testing time comparisons on NIPS collection for nonparametric topic models. 191
6.11 Training and testing time comparisons on OHSUMED collection for nonparametric topic models. 191
6.12 Training and testing time comparisons on Reuters collection for nonparametric topic models. 191
6.13 Training and tesing time comparisons on AQUAINT-1 collection for nonparametric topic models. 192
6.14 The effect of topic Dirichlet parameter on classification for different nonparametric topic models. 201

7.1 Graphical representation of our proposed document classification model. 207
7.2 Per-class distribution over topics in *comp.graphics* class of 20 Newsgroups dataset. 213
7.3 Per-class distribution over topics in Class 5 of OHSUMED-23 dataset. 214
7.4 CPU runtime performance for supervised topic models. 216

8.1 Our document retrieval learning topic model without word order. . . 222
8.2 Second graphical model of our document retrieval learning model where the order of words in queries is relaxed. 225
8.3 Document retrieval learning model with word order in both queries and the documents. 227

9.1 An illustration of different terrains. 243
9.2 An illustration of a readability segment. 251
9.3 Effect of varying the readability weight parameter. 260

10.1 A pie chart showing the approximate time that I have spent in different tasks during my research study. 267

List of Symbols

$[\gamma_n^x]$ A vector of $N \times 1$ which encapsulates the terms weights in the concept space.

α In case of `NTSeg`, α is a $K \times L$ matrix where each row represents the mixing proportion of the word-topics in a segment-topic, in other parametric topic models, it is the prior for the document-topic multinomial distribution. It is assumed as a symmetric distribution in this thesis.

α In case, of the Bayesian nonparametric topic models, it is the concentration parameter.

$\alpha_{y_s z_{si}}$ When we refer to an element in $\alpha_{y_s z_{si}}$, we generally follow this notation paradigm, and it means that it is the z_{si}^{th} component in α_{y_s}.

$\bar{m}_{kl}$ It the sample mean which is computed over all the segments assigned to the segment-topic k.

$\bar{v}_{kl}$ It is the sample variance which is computed over all the segments assigned to the segment-topic k.

β In the parametric and nonparametric topic models, it is the parameter of the prior probability for distribution of the words conditioned on the word-topics.

β In the readability model, it is the parameter controlling the relative contribution between term difficulty and cohesion.

$\boldsymbol{\kappa}$ It is the mean of classifier parameters η

$\boldsymbol{\Theta} = \{\theta^d\}_{d=1}^D$ It represents are topic distributions for all documents.

$\boldsymbol{B} = \{\boldsymbol{b}^d\}_{d=1}^D$ Encodes the word order information.

$\boldsymbol{b}^d$ It represents the following form $\{b_{n,n+1}^d\}_{n=1}^{N^d-1}$.

$\boldsymbol{f}(y,(d,q))$ Represents a vector of features which are designed to be useful for retrieval

$\boldsymbol{f}(y,\overline{\boldsymbol{z}}^d)$ It is a MK-dimensional vector whose elements from $(y-1)$K to yK are $\overline{\boldsymbol{z}}_k^d$ and rest are all 0.

$\boldsymbol{W} = \{\boldsymbol{w}^d, y^d\}_{d=1}^{D}$ The training set.

$\boldsymbol{w}^d = \{w_i^d\}_{n=1}^{N^d}$ Word appearing in the document d

$\boldsymbol{Z} = \{\boldsymbol{z}^d\}_{d=1}^{D}$ Topic assignments to all the words in the corpus.

δ It is the Dirichlet prior of σ

η In case of the document retrieval learning model, it represents the model parameters which are essentially feature weights.

η In supervised topic models, it is a random variable representing the parameter of the classification model.

γ It is the Dirichlet prior of ψ.

γ Prior distribution for ψ.

$\hat{n}_k$ It is the number of segments assigned to the segment-topic k.

$\hat{S}_k$ It is the set of segments assigned to the segment-topics k.

$\mathbb{I}(.)$ It is an indicator function which equals to 1 if the predicate holds else it is 0.

$\mathbf{L}$ It is a matrix of dimension $f \times D$

$\mathbf{R}$ It is a matrix with dimension $W \times f$.

$\mathbf{S}$ It is a $f \times f$ diagonal matrix of singular values.

$\mathbf{U}$ It is a $W \times f$ matrix of left singular vectors.

$\mathbf{V}$ It is a $D \times f$ matrix of right singular vectors.

$\mathbf{V}^T$ It denotes matrix transposition of $\mathbf{V}$.

$\mathbf{w}$ A vector comprising of words.

$\mathbf{x}$ Binary status variable vector. This variable consists of status values of words in sequence for a document.

$\mathbf{y}$ Segment-topic variable vector for a document.

$\mathbf{z}$ Word-topic variable. In case of the `NTSeg` model we explicitly use the term word-topic, because of two levels of topics that our model generates. In general it is synonymous to latent topics or just topics that we will use for the rest of the models.

$\nu(\vec{\Delta}_s, \vec{\Delta}_{s+1})$ It denotes the cosine similarity between the centroids of two clusters to which the segments belong.

Ω The parameter of the Beta distribution.

$\Omega_{z_i^d 1}$ It is one of the shape parameters of the Beta distribution.

$\Omega_{z_i^d 2}$ It is one of the shape parameters of the Beta distribution.

$\overline{\boldsymbol{z}}^d$ It is a L dimensional vector with each element $\overline{z}_k^d = \frac{1}{N^d} \sum_{n=1}^{N^d} \mathbb{I}(z_n^d = k)$.

$\overline{t}_z$ It is the sample mean.

ϕ In both the parametric and nonparametric topic models, it is the parameter of the Multinomial distribution of word conditioned on the word-topics.

ϕ_{zw} It represents the bigram word distribution.

π The parameter of the Binomial distribution.

ψ It is the Bernoulli distribution of the status variable x_i^d with respect to the previous word.

ρ It is the parameter of the Dirichlet prior on the segment-topics.

σ The parameter of the Multinomial distribution with the prior δ.

τ In the `NTSeg` model, it is the mixing proportion of the segment-topics in a document.

τ In the terrain model, it is the average number of terms of all segments in the document.

θ^d In other parametric topic models in this thesis, it is defined as a matrix which contains the document-topic distributions.

$\theta^{(s)}$ In case of `NTSeg` it is the mixing proportion of the word-topics in the text segment s.

$\vec{\Delta}_s$ It denotes the centroid of the cluster in which a segment Q_s exists.

$\vec{l}_j$ It denotes a document vector at column j in $\mathbf{L}$.

$\vec{r}_x$ It is the term vector at row x in matrix $\mathbf{R}$.

ξ^d It represents the slack variable.

ζ_j It denotes the overall cohesion score of document j.

$b_{n,n+1}^d$ It denotes the words at the positions n and $n+1$ in the document d.

$c(w_i^d, d)$ It is the number of times the word w_i^d appears in the document d.

C It is a regularization constant.

c_s^d A binary switching random variable which indicates whether there is a change in segment-topic from one segment to another in sequence of segments in the document.

D The number of documents in the collection.

d A document variable which represents one document.

$f << \min(W, D)$ It is the number of factors in the Singular Value Decomposition (`SVD`).

G A random distribution which is distributed according to a Dirichlet Process (DP).

H In case of the Bayesian nonparametric topic models, it is the base distribution.

J_n In the Normalized Discounted Cumulative Gain (NDCG) formula, it is the normalization constant such that a perfect list gets a score of 1.

K Number of segment-topics for the entire corpus, which is specified by the user in advance.

L Number of word-topics for the entire corpus, which is specified by the user in advance.

$l^d(y)$ It is the loss function for the label y.

M Number of classes considered in the classification problem.

m_{wv} It is the number of times the word v is assigned as the second word of a bigram given the previous word w, and given the same topic of the previous word.

m_{zwv} It is the number of times word v has been assigned to z as the second term of a bigram when the previous word is given.

n In the NDCG formula, it is the length of the ranked list.

N^d Total number of words in the document.

N^q It is the number of words in the query q.

N_s^d Number of n-gram words in the document.

n_{d_0} It is the number of times the switching variable c_s is set of 0 in the document d.

n_{d_1} It is the number of times the switching variable c_s is set of 1 in the document d.

$n_{z_{si}^d}$ It is the number of times a word in segment s of document d is assigned to word-topic z.

n_{zw} It is the number of times that the word w is assigned to the word-topic z as a unigram.

$P_{d_{y_s^d}}$ It is the number of times a segment in the document d has been assigned to the segment-topic y_s^d.

p_{zwt} It denotes the number of times the status variable $x = t$ (0 or 1) in the same topic z as the the previous word w.

Q_s A segment is represented by this variable.

q_{dz} It is the number of times a word is assigned to topic z in document d.

S Number of segments in a document.

s A segment variable.

S_j It is the total number of segments in document j.

s_z^2 It is the biased sample variance of the time-stamps which belong to z.

t_i^d The time-stamp variable of a document.

W Number of words in the vocabulary.

x_i^d Binary status variable. This variable tells us whether two words in sequence i.e. words w_{i-1}^d and w_i^d form a bigram or not in the document d.

x_{si}^d The bigram status variable. This variable tells us whether a word w at position i in segment s of document d, denoted as w_{si}^d, forms a bigram with the word w_{si-1}^d.

y^d It is the class label which takes on one of the values $\mathtt{Y} = \{1, \ldots, M\}$.

y_s^d It is the segment-topic that has been assigned to the paragraph s in document d.

$y_{\neg s}^d$ It is the segment-topic assignments for all the segments except the current segment s.

z_{si}^d It is the word-topic assignment for the word w_{si}^d in segment s of document d.

Dirichlet (α) It represents the Dirichlet distribution with parameter α

r(i) In the NDCG formula, it is the rank label of the i^{th} document in the ranked list.

List of Algorithms

1 Inference algorithm for `NTSeg`. 114

2 Inference algorithm for the `NTOT` model 140

3 Cohesion based on segmentation. 250

List of Abbreviations

`ATM` Author Topic Model.

`BTKM` Bigram Token Model.

`BTM` Bigram Topic Model.

`CHM` Conceptual Hop Model.

`CRF` Chinese Restaurant Franchise.

`CTM` Correlated Topic Model.

`CTM` Compound Topic Model.

`DPMM` Dirichlet Process Mixture Model.

`DTM` Dynamic Topic Model.

`DiscLDA` Discriminative Latent Dirichlet Allocation.

`EM` Expectation-Maximization.

`GD-LDA` Generalized Dirichlet Distribution - Latent Dirichlet Allocation.

`GTM` Group Topic Model.

`HDP` Hierarchical Dirichlet Processes.

`HMM` Hidden Markov Model.

`HPYP` Hierarchical Pitman-Yor Process.

`IDF` Inverse Document Frequency.

`LCA` Latent Class Analysis.

`LDACOL` Latent Dirichlet Allocation-Collocation.

`LDA` Latent Dirichlet Allocation.

LDCC Latent Dirichlet Allocation Co-Clustering.

LDSEG Latent Dirichlet Allocation Segmentation.

LSA Latent Semantic Analysis.

LSI Latent Semantic Indexing.

MedLDA Maximum-Margin Entropy Discrimination Latent Dirichlet Allocation.

NHDP N-gram Hierarchical Dirichlet Process.

NMF Non-negative Matrix Factorization.

NNTM Nonparametric N-gram Topic Model.

NTOT N-gram Topics Over Time.

NTSeg N-gram Topic Segmentation.

PAM Pachinko Allocation Model.

PCA Principal Component Analysis.

PCFGs Probabilistic Context-Free Grammars.

PDLDA Phrase Discovering Latent Dirichlet Allocation.

PYP Pitman-Yor Process.

STTM Sequential Term Transition Model.

SVD Singular Value Decomposition.

SVM Support Vector Machine.

T-BTM Token-Bitopic Model.

TBM Topic-Bigram Model.

TF Term-Frequency.

TNG Topical N-gram.

TOT Topics Over Time.

gMedLDA Gibbs Maximum-Margin Entropy Discrimination Latent Dirichlet Allocation.

kNN k-Nearest Neighbour.

mcLDA Multi-Class Supervised Latent Dirichlet Allocation.

pLSA Probabilistic Latent Semantic Analysis.

pLSI Probabilistic Latent Semantic Indexing.

sLDA Supervised Latent Dirichlet Allocation.

`vMedLDA` Variational Maximum-Margin Entropy Discrimination Latent Dirichlet Allocation.

ARI Automated Readability Index.

BoW bag-of-words.

C-L Coleman-Liau.

CLEF Cross-Language Evaluation Forum.

CRP Chinese Restaurant Process.

DAG Directed Acyclic Graph.

ddCRF Distance Dependent Chinese Restaurant Franchise.

ddCRP Distance Dependent Chinese Restaurant Process.

DP Dirichlet Process.

FAMCLASS Familiarity Classifier.

HDLM Hierarchical Dirichlet Language Model.

HITS Hyperlink-Induced Topic Search.

IID Independent and identically distributed.

IR Information Retrieval.

KL-Divergence Kullback-Leibler Divergence.

MCMC Markov Chain Monte Carlo.

NDCG Normalized Discounted Cumulative Gain.

NLP Natural Language Processing.

PDF Probability Density Function.

SALSA Stochastic Approach for Link-Structure Analysis.

TREC Text Retrieval Conference.

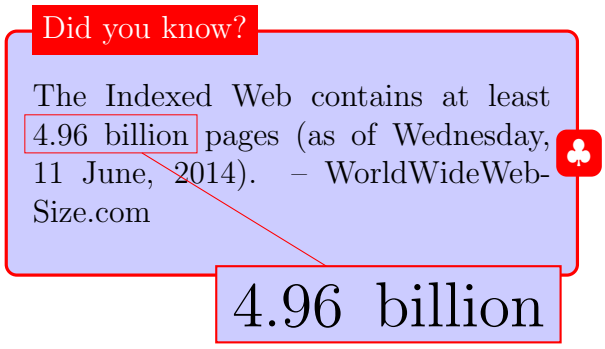
VSM Vector Space Model.

Chapter One

Introduction

1.1 Motivation

Everyday mammoth amount of text data is generated by users on the web. Even if a small subset of this textual data is collected on a computer, the sheer size of it makes it impossible for humans to read them all, and know what each of them is talking about individually. Under such scenarios, we need some automated techniques, which can let a human user know what a particular collection of text documents is talking in totality. Based on the output generated by such an automated system, a user can then select a certain small subset from the entire collection to read rather than manually sift through each of the documents to find the one that interests the user.



In order to get a gist of such voluminous data, probabilistic topic modeling could be a handy tool. A probabilistic topic model takes as input a document collection (preferably all in the same language), and then creates a list of words in each topic which coherently summarizes a given document collection. In Figure 1.1, we depict the input, process and output sequence of an unsupervised bag-of-words probabilistic topic model. By "process" we mean the actual work that goes behind in creating the latent topics along with the topical words. The topic model outputs a list of words in each topic, where the number of topics is pre-defined by the user. Typically, a topic is a probability distribution over words. The summary is in the form of words thus formed in each topic, and words in one topic are closely related to each other. In this way, a user can know what are the topics that pervade the document collection. In Figure 1.2, an example of a list of words formed by a simple topic model, LDA [23], is shown. This result has been obtained from the NIPS document collection. The

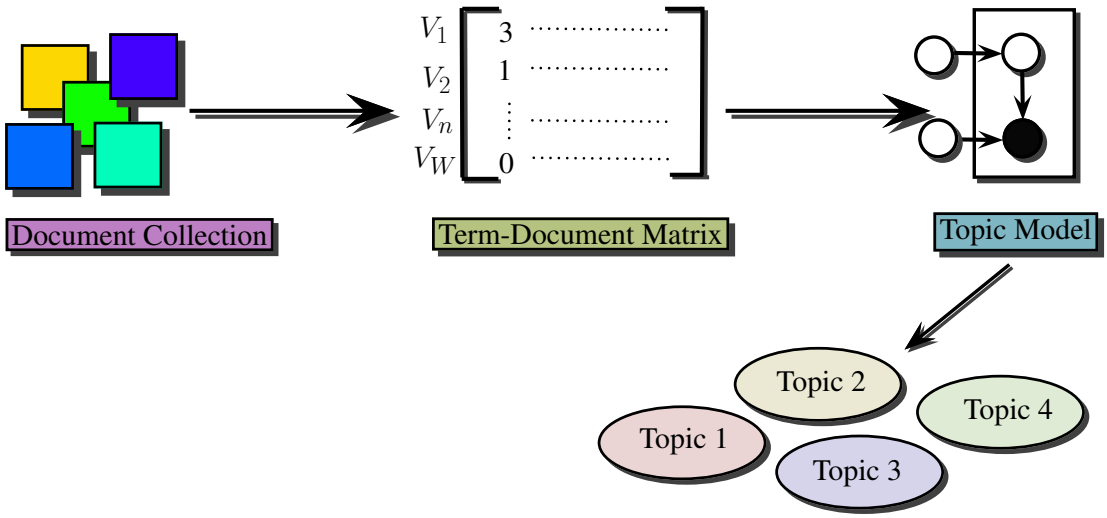


Figure 1.1: The input, process and output of a bag-of-words unsupervised topic model. The user first selects the input type. Inputs could be documents, images, etc. Then the task is to build a co-occurrence matrix of the input data, which falls in the "process" phase. Suppose the input data is a set of text documents. The co-occurrence matrix is a matrix in $\mathbb{R}^{W \times D}$, where W is the vocabulary size and D are the number of documents in the collection. The matrix consists of the number of times a word appears in the document irrespective of its position. This matrix is given as input to a topic model, which then creates a list of words in each topic.

words are arranged according to their decreasing probabilities.

Definition 1. *A text collection, D, is defined as a group of documents.*

Definition 2. *A document, d, is described as a sequence of words which are selected from a vocabulary.*

Definition 3. *A vocabulary, W, is defined as a list of all the unique unigram words in the text collection.*

Definition 4. *A term or a word, w, is defined as one entry (a unigram) in the document which is selected from the vocabulary.*

Definition 5. *A topic or a latent topic is a probability distribution over words in the vocabulary.*

Topic models such as LDA and many other recently proposed models such as [251], [146], [112] have been widely used to find topics in a document collection.

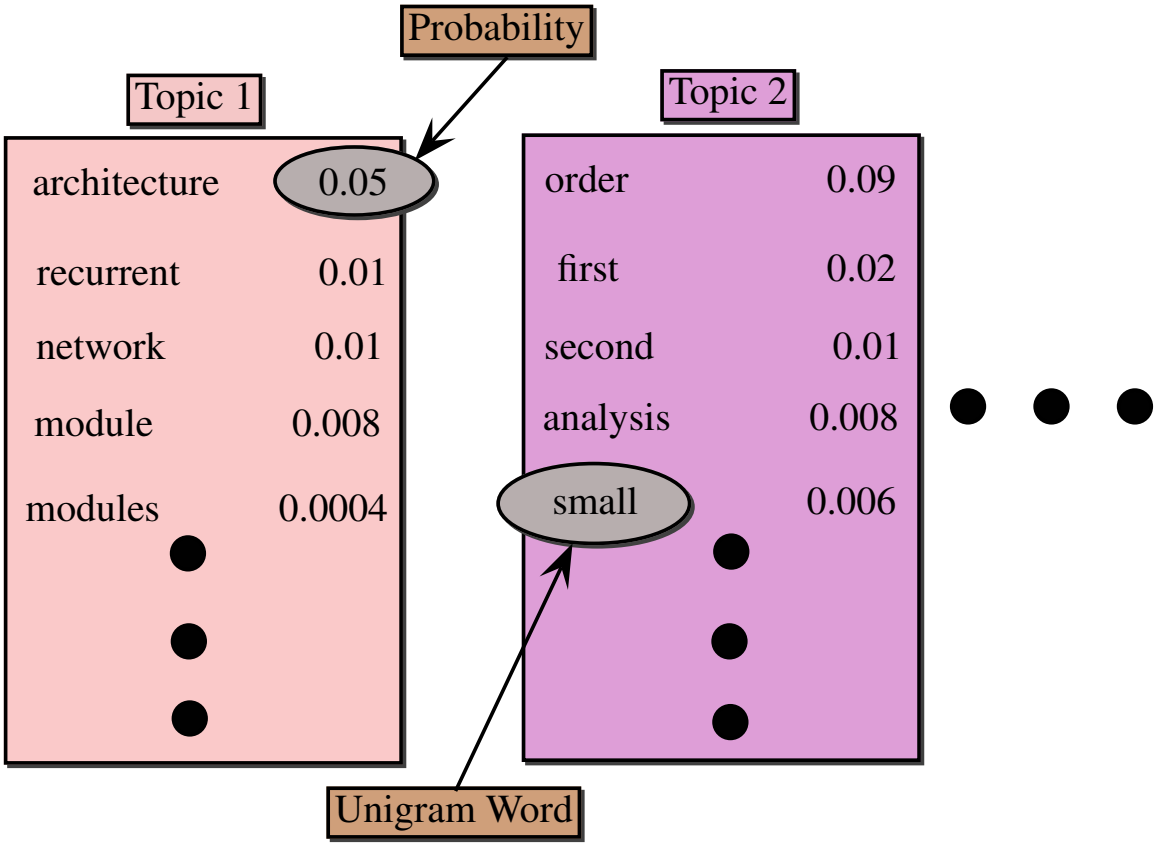


Figure 1.2: An illustration of two topics obtained from the `LDA` model on NIPS document collection. Topic 1 mainly focuses on words whose documents collectively discuss about "neural networks" and Topic 2 focuses on documents which describe about "first order logic".

But the LDA model has been criticized for its bag-of-words assumption [117], as the model does not consider the structural information inherent in the text which could help tap extra knowledge from the text. For example, LDA, due to its bag-of-words assumption, fails to capture a phrase such as "acquired immune deficiency syndrome" which is one of model's shortcoming. It is well known that the bag-of-words assumption is mainly a simplifying assumption to reduce the complexity of the model [166], [115], [117], [88]. Processing documents by keeping the word ordering intact, such as the existing parametric topic models mentioned earlier, does incorporate additional computational burden, nonetheless it gives an upper-hand over traditional bag-of-words topic models [88]. One useful advantage is to discover more interpretable latent topics [261], [117], [116].

Some recent topic models have demonstrated better qualitative and quantitative performance when the bag-of-words assumption is relaxed [115], [136], [1], [166]. In order to address the shortcoming inherent in the LDA model, the authors in [261] introduced the TNG model to find n-gram words in topics. By n-gram we mean a word can be a unigram, a bigram, a trigram word, etc. We have presented an example of different n-grams in Table 1.1. The TNG model has the ability to decide whether to form a unigram or a bigram during the topic discovery process. The TNG model mainly extends the LDACOL model [87] and the Bigram Topic Model (BTM) [252]. All these models advocate that the word order in a document is essential. But one shortcoming of these models is that they lack the ability to consider the document's structure such as paragraphs and sentences. Thus they cannot segment a document into coherent topics. This sometimes becomes essential in tasks such as tackling the word sense disambiguation problem as shown in [88], segmenting news articles and finding topics in each segment [220], topic detection and tracking [265], and a plethora of other tasks [218]. In order to address this limitation in the topic models, we proposed the topic segmentation model called, NTSeg [117], to generate n-gram words and also segment a document into coherent topics.

Unigrams	Bi-grams	Tri-grams
thesis	doctoral thesis	random access memory
computer	computer science	chinese restaurant process
science	full moon	hierarchical dirichlet processes
moon	microsoft windows	latent dirichlet allocation
full	white house	nonparametric bayesian model

Table 1.1: Example of unigram words shown in Column 1, bi-gram words shown in Column 2, and Tri-gram words are shown in Column 3.

LDA is an unsupervised probabilistic topic model which analyzes a high dimensional term space and discovers a low-dimensional latent topic space [23]. Many other proposed variants of the LDA model are unsupervised topic models, and have been employed for tackling text mining problems including document classification [117] and document retrieval [266], [261]. These models can achieve better performance via detecting the latent topic structure and establishing a relationship between the latent topic and the goal of the problem. One limitation of unsupervised topic models for document classification is that the topic model itself does not consider useful side-information, for instance, class labels of documents. Another limitation of topic models for document classification is that the topic models do not exploit the word order structure of the documents. Some works attempt to integrate the class label information into a topic model for solving document classification. For example, Supervised Latent Dirichlet Allocation (sLDA) [21] is one model that captures the class label of a document as a real-valued regression response. Wang et al. [257] proposed Multi-Class Supervised Latent Dirichlet Allocation (mcLDA) which captures discrete labels of documents as a classification response. MedLDA [297] and its variants have shown to improve document classification performance [299], [123]. The MedLDA model incorporates a maximum-margin principle instead of likelihood-driven objective. However, one common limitation of the above models is that they do not make use of the word order structure in text documents that could interact with the class label information for solving the document classification task. Exchangeability assumption in the probabilistic topic models, helps to simplify the model and also reduces the computational complexity [166], [23], but it has several

Figure 1.3: Comical illustration of the case when a person hears about a word which is ambiguous and which does not give much insight to a human being as to what it is actually referring to. For example, in the illustration above, when a word "house" is shown to a human being, the person is in doubt about this word and the context under which it has been used. In contrast, when someone talks about "White House", immediately the residence of the President of the United States in Washington D.C. comes to ones' mind thereby getting rid of any ambiguity that one may have in ones' mind.

de-merits too, for example, words generated in topics are not that insightful [166] and sometimes ambiguous [261]. We show a comical illustration of such ambiguity in Figure 1.3. In addition, the discovered topics are less interpretable to a user [261]. It is quite common to see words such as "house" in topics generated by the LDA model which is a unigram based topic model. However, generating "house" in a topic does not give much meaning to a reader as there is some doubt in the mind as to which "house" the word is referring to. Instead generating a multi-word expression [20] such as "white house" in a topic clears many doubts that the user may have. Therefore, n-gram based topic models such as the LDACOL [87], TNG [261], etc, help generate more interpretable topics. Some works in the topic modeling literature such as [115] (and also a similar model applied to blogs [1]), [261], [252], [136] and many others have shown to perform better than the traditional bag-of-words counterparts in empirical experimental analysis, such as, held-out likelihood computations, information retrieval, text classification, etc.

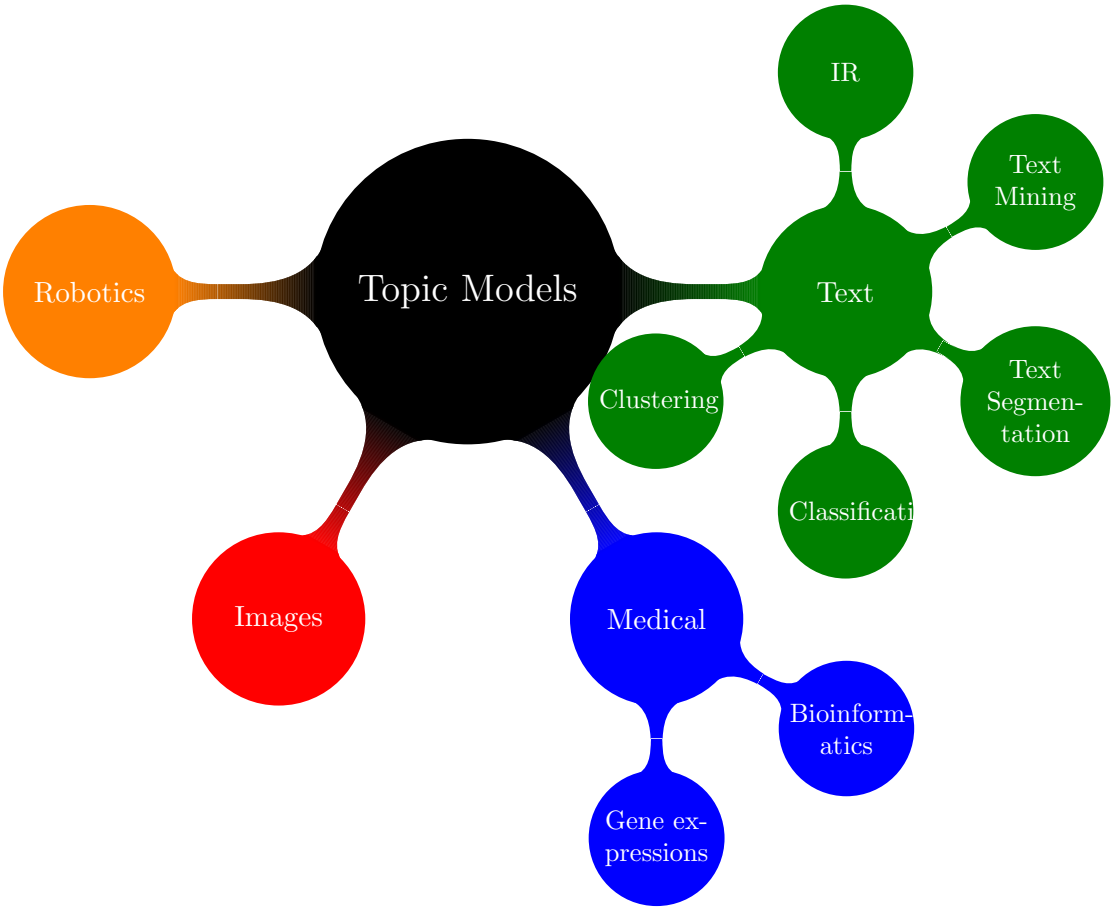


Figure 1.4: A figure showing some of the areas where the topic models have been successfully applied.

1.2 Applications of Topic Models

Topic models have become very popular since they were last introduced. The first latent topic model proposed was the LDA model, which subsequently, has been applied to a wide range of tasks. We show some of the areas where topic models have been applied in Figure 1.4. Topic models have shown immense success in the field of text data, where it has been applied successfully in information retrieval [261], [266], [211], [90], [38], [282], [163], [74], [171], [267], [254], [75], in mining useful information from text [291], [27], [29], [89], [106], [289], social networks [94], [94], [296], in segmenting text into coherent topics [117], [220], in the field of medicine [205], in document classification [295], [162], [34], [200], [181], [294], speech [44], [47],

in document clustering [281], in image data [109], also in Robotics [82], [81], [80], and many other interesting scenarios such as [178], [5], [213], [201], [170], [151].

1.3 Contributions

The goal of this thesis is to contribute new structured topic models for text data. Overall, we propose the following topic models for text data:

- A new unsupervised topic discovery model, called N-gram Topic Segmentation (NTSeg) model, for a collection of text documents. NTSeg maintains the segment structure of the document such as paragraphs and sentences. In addition, it preserves the word order in the document. NTSeg can help capture topical changes in the document from one segment to another. As a result, it can generate two levels of topics of different granularity, namely, segment-topics and word-topics. In addition, it can generate n-gram words in each word-topic. This model is discussed in Chapter 4.

- A new model which can not only consider the local contextual information inherent in the document, but also captures the way in which the topic structure changes over time. By maintaining the word order in the document and capturing phrases in topics can help find words in topics which convey better meaning to the reader. The model uses a continuous distribution over time which is associated with each topic. Topics generate words and observed time-stamp values. The model automatically determines whether to form a unigram or combine with the previous word in each time-stamped document. The main innovation is capturing the evolution of n-gram words in each topic over time. We also present a collapsed Gibbs sampling procedure for efficient posterior inference. This model is described in Chapter 5.

- Considering the limitations of the existing n-gram based topic models, three

new nonparametric n-gram topic models for text data are proposed that can generate insightful n-gram words in topics. Also the proposed models can automatically detect an appropriate number of latent topics from the characteristic of text data. The n-gram nonparametric topic models assume a First-Order Markovian structure on the order of the words in the documents. By introducing a set of binary random variables in the HDP model, and by doing some extra book-keeping during sampling the models can capture topical n-gram words. In addition, corresponding posterior inference schemes for the three models based on the Chinese Restaurant Franchise methods are presented. These models are described in Chapter 6.

- A low-dimensional supervised latent topic model for document classification is presented. Class label information and word order structure are integrated into our supervised topic model enabling more effective interaction among such information for solving document classification. The derivation of the collapsed Gibbs sampler for the proposed model is also presented. This model is described in Chapter 7.

- We then present a new supervised topic model for document retrieval learning problem. Specifically pointwise approach is designed where available relevance assessments and word order structure are also integrated into the topic model itself. The model jointly considers the similarity between the query and the document under a low-dimensional topic space in a maximum-margin framework. One major difference between the proposed model and existing learning-to-rank models is that existing learning-to-rank models do not consider latent topic information in the learning framework. This model is presented in Chapter 8.

- We present the readability models for domain-specific information retrieval that makes use of the latent information generated by the latent concept models to re-rank the search results obtained from a similarity based information retrieval system. Previous approaches have used some external vocabulary or a list of

words to capture domain-specific terms, but we automatically get these words with high scores in the latent concept space. Different schemes are proposed both considering word order and bag-of-words and those schemes are compared. The models are presented in Chapter 9.

1.4 Outline

This thesis is organized as follows. We will first review some of the closely related topic models in Chapter 2. In this chapter, we will present both traditional and current state-of-the-art topic models. In addition, a brief survey about readability methods will also be presented. We will also contrast how the models proposed in this thesis are an improvement over the existing state-of-the-art. We will then present a detailed overview of the related existing state-of-the-art topic models in Chapter 3. This will help build relevant background in order to understand the content in the subsequent sections in this thesis. We will then present a topic segmentation topic model with word order in Chapter 4. In this chapter, we will also present different sets of experimental analysis, and show how the proposed topic segmentation model improves upon the state-of-the-art. In Chapter 5, we will present a topic model for capturing temporal dynamics in data. Specifically, we will capture how topics change over time. In addition, n-gram words that change over time are also captured by the model. We will then show how this model improves upon the existing unigram based topics over time model. In the models proposed in the earlier chapters, we have assumed that the dimension of the latent topic space is pre-defined. But we relax this assumption, and in Chapter 6, present three nonparametric n-gram topic models. We will also present their inference algorithms using the Chinese Restaurant Franchise scheme with Buddy Customers. We will present experimental analysis on

both small and large text collections. The models proposed above do not consider side information or response variable which is available in some datasets. For example, datasets used in supervised document classification problem have some form of a class label associated with them. In Chapter 7, we will present a n-gram supervised topic model which considers annotation information that is available, and which is not considered in the models proposed in the earlier chapters. We will show that we obtain state-of-the-art performance using our proposed supervised topic model. In Chapter 8, we will present a new topic model for document retrieval learning where relevance label is used as a side information. This model conducts learning-to-rank by incorporating the latent topic feature in the topic model. Then we will present the readability methods in the low-dimensional concept space in Chapter 9. Just like the latent topic models, which can be seen as a matrix factorization technique, we use a concept model, Latent Semantic Indexing (LSI), in order to compute the readability of a text document and conduct retrieval of documents based on readability, in addition to relevance. Then we will conclude the thesis by describing some future directions in n-gram probabilistic topic modeling and readability with word order.

Chapter Two

Literature Survey

In this chapter, we will present a detailed literature survey about probabilistic topic models and text readability. We will present both parametric and nonparametric topic models. In addition, we will also present previously proposed works which are closely related to the models proposed in this thesis. Some of the closely related works will be summarized under the following sections.

- **Unsupervised Parametric Topic Modeling with Exchangeability Assumption** - Under this section, we will present literature survey on probabilistic topic models which ignore the word order in the document, and assume that the words in the document are exchangeable i.e. bag-of-words (BoW) assumption. We will present those models which have a pre-defined parameter space that needs to be is explicitly set by the user. We will highlight some of the advantages and shortcomings of these models.

- **Unsupervised Parametric Topic Modeling with Word Order** - In this section, we will present probabilistic topic models that ignore the order of words in a text document. These models capture the document's word order and generate phrasal terms instead of just unigrams. However, these models also require a pre-defined parameter space given by the user.

- **Unsupervised Nonparametric Topic Modeling with Exchangeability** - These topic models automatically find the number of latent topics based on the data characteristics. The complexity of these models grow or shrink with the data characteristics. However, the models that will be discussed under this section are bag-of-words models.

- **Unsupervised Nonparametric Topic Modeling with Word Order** - These topic models find out the number of topics automatically from the data characteristics. The difference is that these models follow the word order in the document and can generate n-gram words. The models in this section will show how order of the words helps improve topic models considerably.

- **Unsupervised Parametric Topic Models for Temporal Data** - Topics rise and fall over time. Some topics are popular at one time, whereas at some other time they are overshadowed by some other topics. The topic models in this section will be related to capturing the temporal dynamics of the data. We will present both unigram and n-gram based topic models. We will also present how the model proposed in this thesis is different from the previously proposed techniques.

- **Supervised Parametric and Nonparametric Topic Models** - The topic models in this section will use an extra side information or response variable to generate more fine-grained latent topics. During model selection, the models make use of a side information which usually comes from some external annotation process, for example, manually annotating data with the help from humans. The topic models that will be discussed under this section will be both parametric and nonparametric. In addition, models which maintain and relax word order will be discussed.

- **Supervised and Unsupervised Readability Prediction Models** - We will present different readability models with and without word order. We will also present our models which make use of the latent concept space for readability prediction. We will present how the models proposed in this thesis are an innovation over the existing techniques. We will also present the limitations inherent in the current readability models.

In addition, we will also contrast about how the models proposed in this thesis are advancements over the state-of-the-art methods.

2.1 Unsupervised Parametric Topic Modeling with Exchangeability Assumption

Parametric topic models assume that the number of parameters are fixed/pre-defined regardless of the sample size, and do not grow with the data [195]. Although it makes the model tractable and computationally efficient, it has some major shortcomings such as the problem of over-fitting (when the number of topics arbitrarily specified are more than what the data can actually accommodate) and under-fitting (when the number of topics arbitrarily specified is less than what the data can actually accommodate). In a parametric topic model, such as LDA [23], [237], [19], [16], the two main inputs are the term-document matrix, and the number of topics supplied by the user. The model then outputs words in each topic with the associated probability (this paradigm has been exemplified graphically in Figures 1.1 and 1.2.). The LDA model posits that documents exhibit multiple topics, and each document is comprised of a mixture of some topics. LDA has been popularly used to discover topics in a document collection. The main assumption of this model is that words in the documents are exchangeable [3], thus it advocates that the order of words in the document is not important. It thus loses an important document's logical order.

Many topic models have been proposed after the seminal work of the LDA model where the bag-of-words model has been used. For example, in order to find topic correlations, Shafiei et al., in [231] described a co-clustering method, known as Latent Dirichlet Allocation Co-Clustering (LDCC), which captures correlation between word-topics and document-topics (or super-topics). The LDCC model is a hierarchical topic model where unigram words are assigned to word-topics, and paragraphs are assigned to the document-topics. The model can also find correlations between word-topics and document-topics. In [161], the authors proposed Pachinko Allocation Model (PAM) where the concept of topic is extended to not only including distributions over words, but also distributions over topics. This model as-

sumes the structure of an arbitrary Directed Acyclic Graph (DAG) in which each leaf is associated with a word and each non-leaf node is a distribution over its children. The interior nodes are distributions over topics called super-topics. Recently, in [35], the authors presented a new model to find correlation among topics in a corpus using the Generalized Dirichlet distribution model instead of the Dirichlet distribution. The model is called Generalized Dirichlet Distribution - Latent Dirichlet Allocation (GD-LDA),

Unigram based topic models have been used in topic segmentation task too. The goal of topic segmentation is to group the segments in the document based on topical changes. A segment could be a sentence or a paragraph. In [183], the authors presented a method for topic segmentation based on topic modeling where the authors used the LDA model to segment texts into coherent topics that assume exchangeability among the words in a document. In [22], the authors described a topic segmentation method by unifying the segmenting Hidden Markov Model (HMM) in [185] and the aspect model in [102]. Recently, in [220], the authors presented TopicTiling based on LDA. Their algorithm is very similar to the TextTiling [97] algorithm, and segments documents using the LDA topic model. Also, in [219], the authors presented methods in which topic models can help segmentation based methods by extending their own TopicTiling model. Similarly, in [230] the authors proposed a topic segmentation based topic model, known as LDSEG, where they assumed that the word order is not important. The authors introduced the notion of topic hierarchy where sentences are assigned to the document-topics and unigrams are assigned to word-topics. The LDSEG model represents documents as a distribution over document-topics or super-topics in such a way that each segment is assigned a super-topic or document-topic, which is then used to choose the parameters of a document independent Dirichlet distribution from which word-topics for the segment is drawn. In order for consecutive segments to have similar word-topic distributions, an additional binary variable per segment encodes whether the document-topic is forced to be the same as that of the previous segment. In [48], the authors proposed a topic model based hierarchical segmentation approach where they assumed that the word order within the segment

is not important, and apply variational Bayesian Expectation-Maximization (EM) procedure for computing the posterior inference. This model has been designed for segmenting the speech data. In [63], the authors proposed a collapsed Gibbs sampler for the topic segmentation problem for a faster posterior inference. They employ a hierarchical Pitman-Yor Process (PYP) to handle hierarchical modeling. In [241], the authors presented a topic segmentation model which does not find topics in a segment. In [46], the authors proposed a subsequence based topic segmentation approach which uses a suffix tree model for representing text, and measures coherence between sentences based on subsequence. Their model maintains the order of the words in each segment, but the model does not find collocations in that text segment.

There are some models which use the LDA model to generate n-gram words, for example, Kim et al., [136] used the LDA model along with the frequent pattern mining approach to capture word order in the document. In their methodology, the authors first mine the frequent patterns from the data. These frequent patterns represent the semantic associations between the units in the collections. The frequent pattern information is then fed to the conventional bag-of-words topic model which can further capture the semantic associations among the frequent patterns. An advantage of this approach is that the methodology is computationally efficient. There are already several computationally efficient frequent pattern mining algorithms to capture such semantic associations. The unigram based topic models such as LDA has also undergone many algorithmic innovations. Several fast LDA algorithms have since been proposed [208], [264] and many others which can scale to large datasets and can also show faster convergence.

The method of Kim et al., [136] also suffers from disadvantages in that one needs a adopt a two-step approach in order capture the word dependencies in the data. This is indeed time consuming in terms of the amount of labour hours, but is also heavily dependent on the quality of the frequent patterns generated from the data. There are several other challenges facing the frequent pattern analysis, for example,

scaling the methods to very large datasets [92] which might generate exponential number of frequent patterns. Then there is a limitation on the number of topics in the LDA model.

In contrast to the frequent pattern mining approach, our proposed topic models are single step approaches that can take advantage of the order of the words in the document, and can capture the semantic associations between the words in the documents. Our methods can also scale to large datasets. One similarity between our work in [117] and the topic correlation models is that our NTSeg model also introduces two levels of topic assignments. For example, word-topics and document-topics as described in Shafiei et al., [230], [231] share the same notion as word-topics and segment-topics described in our proposed approach. We adopt the name segment-topics because known text segments such as paragraphs or sentences are assigned to the segment-topics. However, all the correlation topic models mentioned above assume exchangeability among the words in a document. The importance of capturing n-gram words is that it reduces the ambiguity in the mind of the reader as to what the word is referring to in the correlation graph. For example, presenting the word "networks" in a topic is ambiguous especially for a person who is not a domain expert. In contrast, showing the word "neural networks" in a topic significantly reduces ambiguities. In addition, popular topic models such as the LDA model lacks the capability to capture correlations between the topics of the words in the document.

2.2 Unsupervised Parametric Topic Modeling with Word Order

Capturing word order and thus generating topical n-grams has caught some attention in the past. Wallach [252] proposed the BTM for text data that maintains the order

of the words in the document. The model incorporates the hierarchical Dirichlet language model [172] into the LDA model, and thus captures only bigram words in topics. The model achieves better predictive accuracy than the LDA model, and also solves another problem which is widely discussed in the LDA model where the common words dominate the topics immensely. The model, however, requires the number of topics to be pre-defined by the user.

One limitation of the BTM model is that is always generates bigram words, which at times, might not be intuitive because words do not always exist as bigrams. For example, "white house" may sometimes exist as a bigram, but in some other topic "white" and "house" may exist independently. In order to solve the problem inherent in the BTM model, Griffiths et al., [87] proposed the LDACOL which has the ability to generate both unigram and bigram words. The LDACOL model introduces a new set of binary random variables which indicates the bigram status of a word. An advantage that the model possesses is that it can generate both unigram and bigram words which are more insightful than just unigram words. However, it has a limitation in that only the first word in a bigram has a topic assignment. The model has other shortcomings which were then addressed by the TNG model proposed by Wang et al., [261], where both words in a bigram get the topic assignment. The TNG model has the ability to decide whether to generate a unigram or a bigram for the same two words depending on their nearby context. But it also has some limitations, for example, words in a bigram do not share the same topic assignment.

Lindsey et al., [166] proposed an improvement over the LDA model which generates phrases Phrase Discovering Latent Dirichlet Allocation (PDLDA) that incorporates the Hierarchical Pitman-Yor Process (HPYP) in the LDA model. The model is well suited for finding topical phrases, and has shown to generate more insightful phrases than the TNG or LDACOL models. Technical descriptions about the PYP and HPYP, and why they are suited for text data can be found in [244]. The model proposed by Lindsey et al., [166] indeed gives a plausible solution to many problems

that are inherent in the n-gram topic models proposed so far in the literature, such as, words in the topical phrase share the same probability mass. The model outperforms closely related models in phrase intrusion test. The model however suffers from some severe drawbacks. The main concern is that the model cannot scale to accommodate large text collections due to the HPYP model incorporated in the topic model. Bartlett et al., [9] have found out that the HPYP model is impractical for large datasets until some refinements are done. They introduced such a refinement and tackled the problem of sequence memoizer. But for capturing topical phrases, such solutions might not work. In [255] and [67], the authors presented topical phrase extraction method and constructed a topical hierarchy of phrases. Their model also performs phrase ranking along with topical tree construction using recursive clustering. Their focus is primarily on topical phrase extraction and constructing topic trees from the extracted phrases.

Recently, in [8], the authors have presented three models for sequential data. The authors mainly designed their models for solving the problems related to sequential data such as web data logs, customer purchase history, and other related problem domains where sequence matters very much. In one of the models, called as the Bigram Token Model (BTKM), the authors assumed a first-order Markovian assumption on the order of the words and captured word dependencies in sequence. In the Topic-Bigram Model (TBM), the authors capture dependencies between the topics in sequence. The Token-Bitopic Model (T-BTM) does not consider word dependencies rather it considers topic dependencies in sequence where a word is not only generated by the current topic, but also from the previous topic. One problem with these models is that in the text data the model will always generate bigram in the case of Token-Bigram model, just like the BTM. Consequently, these models might not be a very plausible models to consider. The models appears to be good for other applications as mentioned in the paper, but it is not well suited for text data where words appear as unigram, bigrams, and even higher order n-grams.

Lau et al., [150] presented a study investigating whether word collocations can help improve topic models. They also studied the impact of incorporating word collocations in topic models. Their model does not automatically generate word collocations, rather the collocation discovery is done during the preprocessing stage, thus making it a two-step procedure similar to the one described in [136]. Instead of considering unigrams as the atomic input units to a topic model, the authors instead consider such collocations as an input to a topic model. The quality of the generated topics will primarily depend on the quality of the collocations formed.

In Johri et al., [129], the authors introduced a multi-word enhanced author-topic model for text data. The model can cluster authors with similar expertise and interests. The main advantage of the model is that it can find multi-word expressions from the data instead of unigrams. The model also retains the properties of the unigram based models such as simplicity and computational complexity. However, the model is designed for author and their interest retrieval. It also has a disadvantage that one has to explicitly pre-define the number of latent topics a priori. In [280], the authors presented another n-gram topic model that is well catered for news thread extraction. This model maintains a background distribution over the corpus, and also a multinomial distribution over the hidden news threads. The model closely resembles the TNG model, but incorporates a background distribution which generates the most common words across news threads. The model generates better interpretable n-gram words in topics as compared with the traditional unigram based models. Lin et al., [165] proposed a model that utilizes both the advantages from the topic model and also n-gram language model. The basic intuition of the model is that the topic model is able to capture long range semantic relatedness which the n-gram language model cannot, but the n-gram language model can capture short range semantic relatedness. By combining the advantages from the approaches the authors obtained better probability distribution of a word based on its context using the LDA inference procedure.

There are many differences between the models proposed above, and our proposed topic models in this thesis. Our nonparametric topic models in Chapter 6 can automatically detect an appropriate number of latent topics from the characteristic of text data. Our n-gram nonparametric models assume a First-Order Markovian structure on the order of the words in the documents. By introducing a set of binary random variables in the HDP model, and by doing some extra book-keeping during sampling we can capture topical n-gram words. We also present the corresponding posterior inference schemes for the two models based on the CRF methods. Our model can also scale to large document collections. In Chapter 5, our n-gram topic model can capture n-gram words over time which none of the above models can accomplish. Our NTSeg described in Chapter 4 can segment a document into topics, and can also find n-gram topical words in each segment. Also, our NTSeg model gives the same topic assignment to all words in an n-gram unlike TNG.

2.3 Unsupervised Nonparametric Topic Modeling with Exchangeability

Nonparametric topic models are the models on an infinite-dimensional parameter space where the complexity of the model grows with respect to the sample size. It is not true to say that these models do not have parameters. The models indeed contain parameters, but those parameters are not bounded and can grow based on the complexity of the data. In such topic models, the main input to the model is the term-document matrix. The advantage of the nonparametric method is that one does not need to pre-define the number of latent topics, but these models are highly sensitive to the concentration parameters.

The seminal nonparametric topic model is the HDP model proposed by Teh et al., [247]. Although the model can be applied to a variety of tasks [246], it can also be

used in topic modeling where the number of topics is automatically determined by the data characteristics. Our description will be primarily based on its application in the domain of topic modeling. The model assumes that words in the documents are exchangeable and thus cannot capture short-range word dependencies. There are in fact several extensions proposed to the HDP model, for example, [56] where the author captures both syntax and topical words in one model itself. Nguyen et al., [192] apply for the Bayesian nonparametrics in segmenting speech discourse. Fox et al., [71] use Bayesian nonparametrics to speaker diarization task.

In contrast to the above methods, our nonparametric topic models maintain order of words in the document. This helps them to capture both short and long term co-occurrences. Our models also generate more interpretable latent topics.

2.4 Unsupervised Nonparametric Topic Modeling with Word Order

Considering the order of the words in case of the Bayesian nonparametrics is beginning to attract some attention recently. Goldwater et al., [83] presented two nonparametric models for word segmentation. Observing that ordering of the words could play a dominant role, Goldwater et al., extended the unigram based model to a bigram based model called the "Bigram HDP" model, which maintains the ordering in text. The model closely resembles the HPYP model and can capture dependencies. The model does not find topics but it is well suited for the word segmentation task. The model has some shortcomings and subsequently some corrections were proposed in [24]. Further refinements were then proposed in [84]. In [134], the author proposed a supervised topic model considering word order. The model adopts a nonparametric topic modeling approach, but makes use of an extra supervised signal during inference procedure. The model is different from our nonparametric topic models in that

ours is an unsupervised topic model, and does not need human labeled annotated data.

Johnson [128] presented a connection between Probabilistic Context-Free Grammars (PCFGs) and the LDA model. The findings of Johnson suggests that the inference scheme that is used for PCFGs can also be used for the LDA model. Subsequently by extending the model to incorporate nonparametric adaptor grammars, Johnson is able to discover word collocations instead of just unigrams using the extended LDA model with adaptor grammars. However, a disadvantage of Johnson's method is that he adopts a two-stage approach towards collocation discovery whereas our nonparametric topic modeling approach is single step. In [57], the author introduced a nonparametric model that can extract phrasal terms based on the mutual rank relation. This model first extracts phrases, and subsequently ranks them. It employs a heuristic measure for the identification of phrasal terms. The model proposed is mainly a phrase ranking model and is based on heuristics. Our nonparametric methods are based on a principled inference schemes. In [198], the authors introduced the notion of extension pattern, which is a formalization of the idea of extending lexical association measures defined for bigrams. In [284], the authors presented a Bayesian nonparametric model for symbolic chord sequences. Their model is designed to handle n-grams in chord sequences for music information retrieval.

2.5 Unsupervised Parametric Topic Models for Temporal Data

Blei et al., [18] introduced Dynamic Topic Model (DTM) to capture the way topics evolve over time. They assumed that topics in one year are dependent on the topics of the previous year, which is a discrete distribution over time assumption. The problem with time discretization is that one needs to explicitly select an appropriate time slice

value. In contrast, our model in Chapter 5 assumes a continuous distribution over time. Wang et al., [256] extended [18] and proposed a continuous time dynamic topic model where they used Brownian motion to model the sequential collection of documents, but they adopted a bag-of-words approach. The authors in [143] employed a Compound Topic Model (CTM) to model the temporal dependencies in data, but assumed a discrete distribution over time. In [86] the authors studied an ordering of documents in time and then slicing them into discrete time intervals to capture the temporal nature in data. In Group Topic Model (GTM) [262] the authors divided the UN voting records into segments and the group topic model was fit to each segment which is again a discrete time assumption. Swan et al., [240] described a method to capture time related information in a news corpus. The model constructs "overview timelines" of a set of news stories based on discrete time assumption. Jo et al., [125] proposed a method to present a topics over time model where they conceptualized a topic as a quantized unit of evolutionary change in content and then found temporal characteristics in a corpus. This helped build topic chronology which again selects the time slice discretely. Yin et al., [283] proposed a latent periodic topic analysis, a variant of the LDA model, where their model exploits periodicity based on co-occurrence. This results in finding periodic topics.

In [133], the author introduced a trend analysis model to capture how topics evolve over time. The trend class has a probability distribution over temporal words and a continuous distribution over time. But the author adopts a bag-of-words approach. In [209], the authors presented a hierarchical Bayesian model to capture the temporal nature inherent in the data. Their model infers a change in the topic mixture weights as a function of time. The documents are each characterized by a topic where topics are drawn from a mixture model. A major difference between their work and our n-gram temporal topic model is that they measure a change in the topic mixture weights over time. In contrast, we measure how topics evolve over time. In [193], the authors presented a continuous time model where the model is a Bayesian network. This model uses a Markovian assumption which our model

does not presume. Kleinberg [141], presented a burst and activity model that uses a probabilistic infinite automaton. The model assumes a Markov order in words with the aim of finding temporal patterns. The model operates on only one word at a time, but our model makes use of the word co-occurrence patterns with an ability to form phrases. In [25], the authors proposed a segmented topic model which is based on the Author Topic Model (ATM) [222] to integrate the temporal structure in the corpus into a topic model but they assumed bag-of-words in each segment. Hong et al., [107] introduced a topic model where they incorporated the volume of terms into the temporal dynamics of topics. The authors combined state-space models with term volumes in a supervised method. In contrast, our n-gram temporal topic model model requires no human supervision. In [174], the authors presented a Bayesian topics over time model and stated that the original TOT model [260] is likely to overfit to the time-stamp data and they applied a prior distribution to the Beta distribution in order to tackle this issue. A limitation of their model is that it is a highly complex graphical model which only considers unigrams in topics. A limitation with all the models proposed above is that all assume independence among the words in documents, and hence cannot form phrases in topics. This results in sub-optimal results as far as word discovery in each topic is concerned because many words may be ambiguous.

2.6 Supervised Parametric and Nonparametric Topic Models

Unsupervised and supervised topic models [23], [21], [259] have been used for document classification. An advantage that supervised topic models have over unsupervised ones is that supervised topic models consider the available side-information as response variables in the topic model itself. This helps discover more predictive low dimensional representation of the data for better classification [297]. Blei

et al., proposed the sLDA [21] model which captures real-valued document rating as a regression response. The model relies upon a maximum-likelihood based mechanism for parameter estimation. Wang et al., [257] proposed mcLDA which directly captures discrete labels of documents as a classification response. The Discriminative Latent Dirichlet Allocation (DiscLDA) [147] also performs classification in a different mechanism than sLDA. Different from the above models, Zhu et al. [297], [298] proposed maximum entropy discrimination LDA model known as MedLDA that directly minimizes a margin based loss derived from an expected prediction rule. The MedLDA model uses a variational inference method for parameter estimation. Subsequently, Markov Chain Monte Carlo (MCMC) techniques were proposed in [299], [301], [123], [300]. In [214], the authors proposed a supervised topic model which jointly models available tag-labels by defining a one-to-one correspondence between latent topics and user-tag information. This allows their model to directly learn word-tag correspondences in the topic model itself. What has not been studied in supervised topic modeling is the role that the word order structure in the text document that could play along with the side information in document classification task. Our proposed supervised topic model falls in the class of parametric topic models where the number of latent topics has to be supplied by the user, but recently, Kawamae [134] presented a nonparametric supervised n-gram topic model for phrase extraction which takes the advantage of labels during training process. It places a PYP prior over words and extends the CRF scheme for automatically determining the number of topics. One of the basic differences between Kawamae's and our model is that ours is a parametric model while Kawamae's model is nonparametric. Although one might argue that nonparametric topic models are advantageous than parametric models as the former can automatically find the number of latent topics based on the data characteristics, but the number of topics is highly tied to the concentration parameters. One can adopt hyperparameter optimization techniques which could be computationally expensive, and impractical especially for large datasets. Also it cannot perform document retrieval learning as in our model. Moreover, in [9] it has been stated that nonparametric models with PYP priors cannot scale to large

scale datasets. There are other proposed supervised nonparametric topic modeling approaches such as [196], [238], [149], [273], [164]. These models too cannot perform document retrieval learning task. In addition, such nonparametric topic models are computationally very expensive.

Unsupervised topic models have also been used to perform document classification. As mentioned above, they do not make use of the available side-information in the topic model itself. The LDA model is one example and it achieves better performance than that of Support Vector Machine (SVM) [286]. Our NTSeg model, [117] is inspired by the BTM [252]. It relaxes the bag-of-words assumption, and generates collocations just like the LDACOL [87]. In [223], the authors showed a model that maintains the order of words in documents helps achieve better classification results than the state-of-the-art topic models.

Learning-to-rank models have been extensively investigated, and it can be categorized into pointwise, pairwise, and listwise approaches [167]. One early work used some bag-of-features in training a SVM model in order to conduct document retrieval learning which can be regarded as a pointwise approach for learning-to-rank [189]. This approach predicts the binary relevance assessment. Documents are then ranked based on the confidence scores given by the discriminative classifier. Subsequently other discriminative learning-to-rank models have been proposed such as those which handle multi-class relevance assessments [33], [160]. Many state-of-the-art learning-to-rank models have been proposed recently. For example, Wei et. al [76] recently presented a listwise learning-to-rank model, a novel semi-supervised rank learning model which is extended to an adaptive ranker to domains where no training data is available. In [148], the authors presented a sparse learning-to-rank model for information retrieval. Dang et al. [55] proposed a two-stage learning-to-rank framework to address the problem of sub-optimal ranking when many relevant documents are excluded from the ranking list using bag-of-words retrieval models. However, a major difference between these learning-to-rank models and our proposed document

retrieval learning model in Chapter 8 is that our model considers the latent topic information unified within a discriminative framework.

Our graphical model in Chapter 8 shares some resemblance with the graphical models described in [42], [243]. However the model in [42] cannot perform document retrieval learning. Different from [243], our model maintains word order structure, and jointly considers documents and queries for document retrieval learning.

2.7 Supervised and Unsupervised Readability Prediction Models

Unsupervised heuristic readability methods: Much research has been done in measuring the reading level of text [212]. A detailed description about important heuristic readability methods such as Dale-Chall [54], ARI [228], SMOG [176], Coleman-Liau [50] etc, can be found in [64]. These methods compute the vocabulary difficulty of a textual discourse. Their readability prediction is based on computing the number of syllables in a term, number of characters etc, which are the surface level features of text. Heuristic readability methods consist of two components linearly combined into a single formula. The components are - syntactic and semantic. The syntactic component of the readability methods capture sentence length, word length, etc. The semantic component computes the number of syllables and polysyllables, etc. They operate on the assumption that if the sentences are long, then the prime audience for whom the document is meant for are experts in the field. In addition, if the choice of the terms in the document is such that most of them have low syllable counts, then the individual terms are simple for the reader. We present some of the popular readability formulae below, and highlight their semantic and syntactic components:

The Flesch reading ease score is given by the following formula:

$$206.835 - 1.015 \times \underbrace{\frac{\text{Number of words}}{\text{Number of sentences}}}_{\text{Syntactic component}} - 84.6 \times \underbrace{\frac{\text{Number of syllables}}{\text{Number of words}}}_{\text{Semantic component}} \tag{2.1}$$

The Flesch-Kincaid reading ease formula is given by:

$$0.39 \times \underbrace{\frac{\text{Number of words}}{\text{Number of sentences}}}_{\text{Syntactic component}} + 11.8 \times \underbrace{\frac{\text{Number of syllables}}{\text{Number of words}}}_{\text{Semantic component}} - 15.59 \tag{2.2}$$

The Gunning-Fog reading ease formula is given below. In the formula, poly-syllables are words which consist of three or more syllables.

$$0.4\Big(\underbrace{\frac{\text{Number of words}}{\text{Number of sentences}}}_{\text{Syntactic component}} + 100 \times \underbrace{\frac{\text{Number of poly-syllables}}{\text{Number of words}}}_{\text{Semantic component}} \Big) \tag{2.3}$$

The Automated Readability Index (ARI) readability formula is as follows:

$$4.71 \times \underbrace{\frac{\text{Number of characters}}{\text{Number of words}}}_{\text{Syntactic component}} + 0.5 \times \underbrace{\frac{\text{Number of words}}{\text{Number of sentences}}}_{\text{Syntactic component}} - 21.43 \tag{2.4}$$

The SMOG readability formula is as follows:

$$1.043 \times \Big(30 \times \underbrace{\frac{\text{Number of poly-syllables}}{\text{Number of sentences}}}_{\text{Semantic component}} \Big)^{\frac{1}{2}} + 3.1291 \tag{2.5}$$

Let L be the number of letters per 100 words. S sentence per 100 words. The

Coleman-Liau readability formula is as follows:

$$0.0588 \times L - 0.296 \times S - 15.8 \tag{2.6}$$

These methods have long been in existence and still remain a dominant tool for computing the reading difficulty of traditional documents. In fact, many popular word processing packages use them today. However, readability methods tend to perform poorly on domain-specific texts [279] and web pages [52]. There are other shortcomings [30] which undermine their importance. In [187], the authors described an unsupervised method to re-rank the search results of a web search engine in descending order of their comprehensibility using the Japanese Wikipedia, but they failed to address the shortcomings in the readability formulae. Readability prediction has not only remained constrained to English texts, recently in [111], the authors address the problem in Bangla texts.

Why readability methods underperform on domain-specific documents? Consider a short sentence, "In its simplest form, a star network consists of one central switch, hub or computer, which acts as a conduit to transmit messages." The Flesch reading ease [139] score for this sentence is 62.11, which according to the score is not a difficult sentence. However, the sentence carries a deep technical meaning which requires domain-specific knowledge for proper comprehension. Terms such as "star", "network", and "switch" are domain-specific terms in this example but the readability formula has detected them as easy due to the surface level features.

Domain-Specific Readability Methods: To address the shortcomings inherent in the heuristic readability methods, Yan et al., [279] proposed concept based readability ranking method where they have used a domain-specific ontology to capture the domain-specific terms in a document. Their method has a serious drawback in that it requires an ontology for every domain. The authors have only shown the application of their method in one domain. In [137], the authors described concept

readability method in the medical domain. They have used average term and concept familiarity scores from the OAC CHV knowledge base to compute the difficulty of terms and concepts. Zhao et al. [293] presented domain-specific iterative readability method based on grade levels. Their method is influenced by two popular web link structure based algorithms which are Hyperlink-Induced Topic Search (HITS) [142] and Stochastic Approach for Link-Structure Analysis (SALSA) [153]. A limitation of their approach is that they need some seed concepts to initialize their algorithm. This can sometimes be cumbersome as one has to search for a lexicon for every domain. In [188], the authors used Wikipedia to build a list of some technical terms. In contrast, our proposed framework in this paper does not require an ontology or seed concepts, which can be regarded as a major innovation. We have proposed some heuristic terrain models in the LSI space [118], [120], [119], [121], and computed the technical difficulty of text documents and re-ranked the results obtained from a general purpose IR system. A limitation of the terrain models is that they cannot capture n-grams such as *random access memory* etc. Moreover, they lack a solid theoretical foundation. In [122], we presented a document readability and ranking model for text documents which maintains word order. We developed a novel framework to capture suitable n-gram fragments in a domain-specific document by optimizing n-gram fragment sequence connections and taking into account n-gram fragment specificity and cohesion. Also, our method does not require a domain-specific knowledge base. In [43] authors also used LSI method to compute word difficulty.

Supervised Methods for Readability: Although our proposed readability frameworks are completely unsupervised, some supervised methods for computing the reading difficulty of text have been proposed [72]. Supervised learning approach for readability can be considered as a classification problem. In [168], the authors have used SVM [250] for recognizing the reading levels of texts from user queries. They have used syntactic and vocabulary based features to train the classifier. Language modeling has been applied to readability [52] where the authors described

a smoothed unigram model for computing the readability of text documents such as web pages. In [234], the authors also used unigram language model to predict readability. Topic familiarity is different from traditional general readability [145], where the authors studied re-ranking of a search engine result based on familiarity. They also studied the importance of stopwords in their familiarity classifier (FAMCLASS). In [155], classification of health related documents into three levels, namely, Beginner, Intermediate and Advanced is discussed. The authors achieved high classification accuracy using their classifier. In [227], [197] the authors combined word level features with other textual features. They have used SVM together with several word level features to classify documents based on readability. In [98], the authors introduced a k-Nearest Neighbour (kNN) classifier based on grammatical features such as sentence length and the patterns of the parse tree. Bendersky et al., [12] used several features including readability to improve relevance ranking of the web search results.

Readability is a relative measure [249]. In order to cater to the results on an individual user basis, methods using query log analysis have been proposed. Search engine query log mining and building individual user profile classifier can also help to solve the problem to some extent as done in [51], [242], [138]. But this requires confidential and proprietary query log data with private user session details [235]. Many users might not want their sessions to be recorded or used due to privacy concerns [130].

Readability has also been studied in computational linguistics [154]. In [132], the authors used several linguistic and language model features to build a classifier to predict readability of texts. Language model features were found out to be important to their classifier. Pitler et al., [202] used several textual features in their classifier. Their result shows that word features and average sentence length are strong predictors but the strongest ones are the discourse features. One major limitation of the supervised methods is that one needs a large amount of expensive annotated

data [131]. Language modeling approaches cannot capture domain-specific concepts in a domain [293]. In contrast, our proposed readability methods do not need any annotated data.

Our model completely digresses from the past readability approaches, and uses a conceptual model to compute the readability score of a document. Our model first finds a low-dimensional concept space of the original vector space, and then we use this space and word order in the document to compute the readability score of a document. In our model, readability is computed for every word, and the readability scores for the same word varies from one document to the other. It is so because one document might contain more readable usage of the term, where the same term might have been used in a context which are for domain-experts.

Many psychologists have conducted user studies to investigate the nature of texts and features that make them difficult. In [177], the author stated that texts which make the learning path of the reader simple is ideal for an average or below average reader. Experts tend to find documents which are technically sound and have many difficult concepts so that they can further build upon their existing inventory of knowledge. According to the theory devised by Kintsch [140], there are three levels of cognitive representation. These levels lead a reader to comprehend a piece of discourse. The first level is the source code, second is the text-base and third situation model. Organization of the words into sentences constitutes the source code. The surface meaning of the clauses present in the source code constitutes the text-base. The situation model is the mental model that the user builds in her brain using the background knowledge. If the concepts present in a document are semantically related, it indicates that the document constrains itself in one topic only [278]. An information theoretic method for computing the semantic relatedness among the pairs of concepts has been studied in [217]. An important notion that sprouts from the semantic relatedness is the concept of cohesion in texts. Coherence and cohesion

[69], [37] are another two important concepts which have been extensively studied to measure text comprehension. Document cohesion is a state or quality that the elements of a text "tend to hang together" [184]. Texts normally exhibit varying degrees of cohesion [91]. The start of the text will not be cohesive with the later sections of the text [91]. This finding forms an integral backbone in our work and is the main reason why we feel maintaining the term order in the document is essential.

Chapter Three

Background

Chapter Summary

In this chapter, we will review some of the popular latent semantic and probabilistic topic models as the central frameworks for the remainder of the thesis. This section will acquaint the reader with some simple conceptual models proposed earlier, and subsequently the content will digress to the probabilistic topic models. The models described in this thesis, in general, perform what is known as clustering of data. Although the clustering models described in this thesis perform clustering based on dimension reduction, the idea stems from some simple clustering techniques.

3.1 Cluster Analysis

The task of clustering is to divide the data into groups or clusters which bear some meaningful closely related units [113], [93], [206], [79]. Clustering mechanism has been applied in several fields such as Medical science [66], [159], text mining [2], [6], Astronomy [232], [224], databases [285], [276], etc. In Figure 3.1, we present a diagrammatic overview of clustering of data points in some space. In the figure, we can observe that initially some points are in some space, but this space gives us no information about what these points describe. The figure on the right shows the result of clustering these data points where points which belong to one group are clustered together (denoted with the same colour). Therefore, just by looking at some points, one can easily say what those points in this space describe. In general, it would be very difficult to say when the points are not clustered. This is the prime motivation for clustering of points. In Figure 3.2, we also show another example of a clustering phenomenon which is cast as a mixture model where points belong to certain cluster with some weight.

The two kinds of clustering mechanisms in which we will be highly interested are

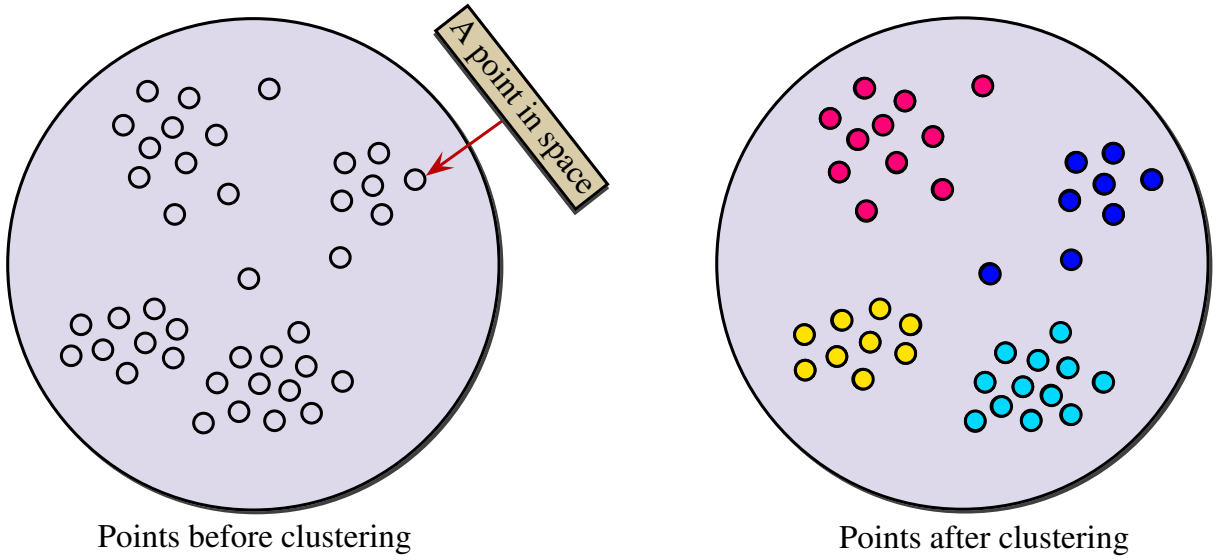


Figure 3.1: An illustration of cluster analysis. The figure on the left shows points scattered in some space which is denoted by a big circle. The figure on the right shows the output generated by a clustering algorithm that groups the set of points based on some rule. Points inside the big circle which are coloured in the same colour belong to one cluster.

hierarchical [96] and partitional clustering. In hierarchical clustering, clusters tend to have subclusters. So there is sharing among the elements in the clusters, whereas partitional clustering imposes a hard assignment of the data points or elements in a cluster, and the elements are not shared among clusters. In order to exemplify the idea further, we depict the notions of partitional and hierarchical clustering in Figure 3.3. In the figure, we see that in case of hierarchical clustering the elements in the cluster are shared. Child elements are shared with the parents. In contrast, in the partitional clustering mechanism, the elements in each cluster are distinct. In the figure, elements with the same colour belong to one cluster and share common properties with the other elements in the same cluster.

Covering everything regarding clustering and types of clusters with some latest state-of-the-art clustering algorithms is out of scope of this thesis, but interested readers are requested to consult few works which summarize state-of-the-art clustering algorithms. Few interesting and comprehensive ones are [26], [114], [186].

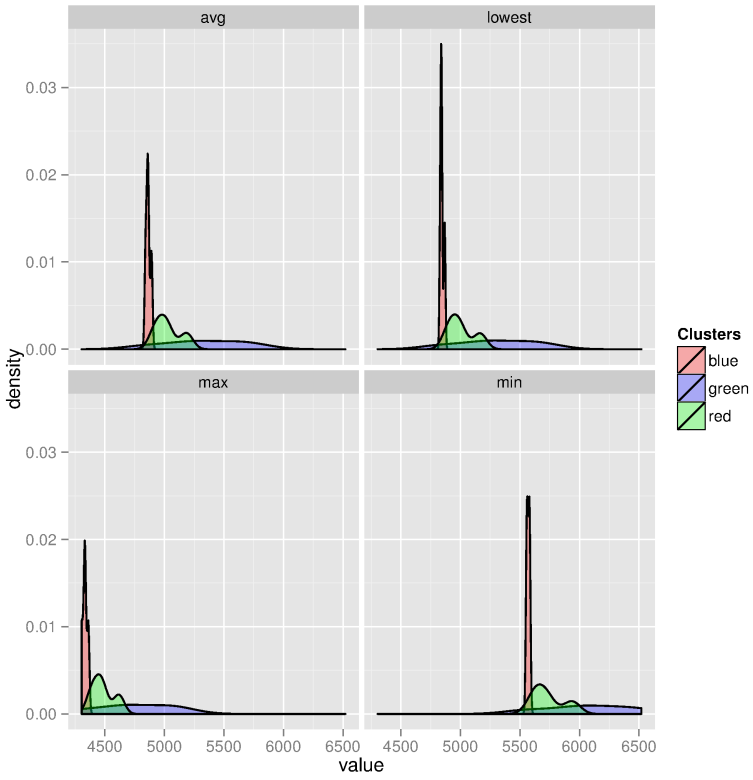


Figure 3.2: Another visualization of a clustering problem. The plot shows how some of the components are mixed with other components. This phenomenon is sometimes called soft clustering.

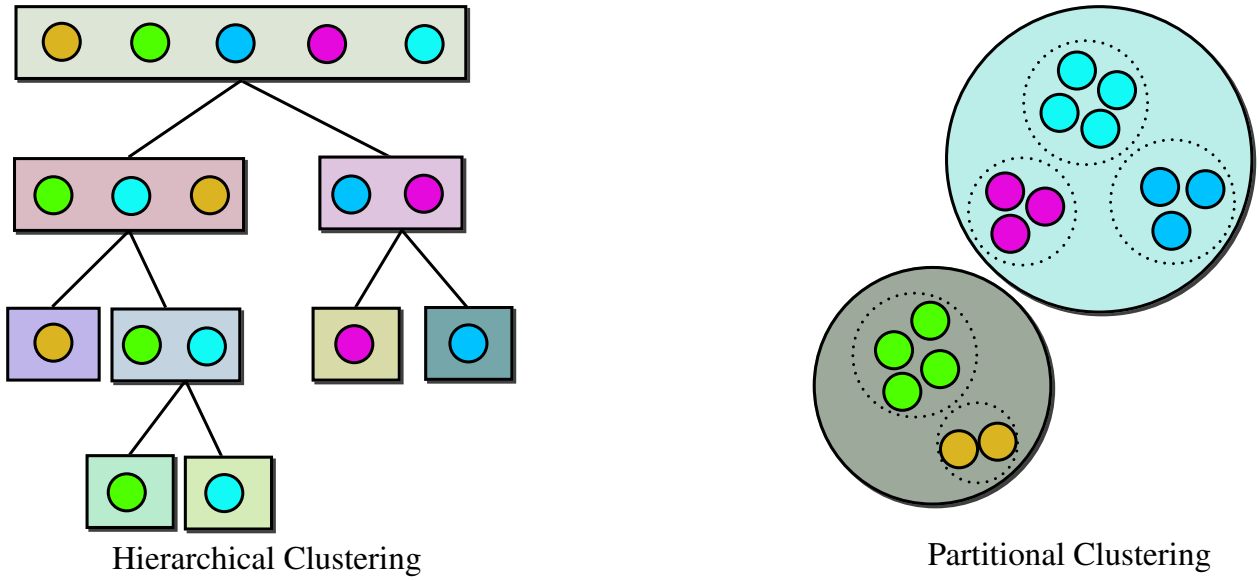


Figure 3.3: A figure illustrating the hierarchical and partitional clustering mechanisms.

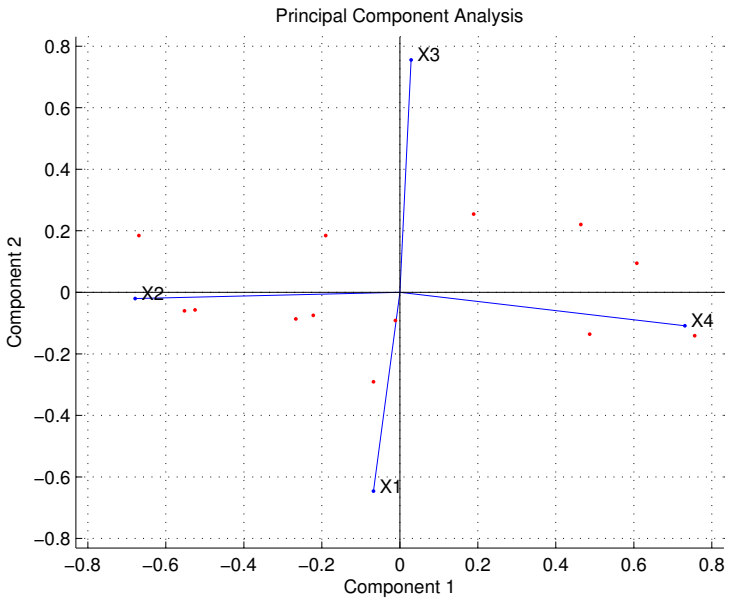


Figure 3.4: A figure showing the principal components.

3.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique of identifying the hidden patterns in the data. When we know about such patterns, then we can express which patterns are similar and which are dissimilar. An advantage of the PCA paradigm is that it helps us focus on the "principal components" i.e. factors which are of prime interest. This is accomplished by reducing the number of dimensions of data without losing much information that the data describes.

In Figure 3.4, we depict the idea behind principal component analysis. We observe that the plot shows some principal components computed from of all the components in the high-dimensional data. The principal components point out some of the main components in the data that best describes the nature of the underlying information in the data.

Models such as LSI, LSA, etc that we shall discuss later in this section, compute the principal components of data and reduce the high-dimensional space to a low-

dimension. This process in turn brings of some important latent classes that is largely hidden in the high-dimensional vector space.

In this section, one popular principal component analysis method known as SVD [85] will be described that forms the basis of one of our models which we shall discuss later.

3.2.1 Singular Value Decomposition (SVD)

Let $\mathbf{A} = W \times D$ be a matrix. In SVD, this matrix can be factored as[1]:

$$\mathbf{A} = \mathbf{U} \times \mathbf{\Sigma} \times \mathbf{V}^{\mathrm{T}} \tag{3.1}$$

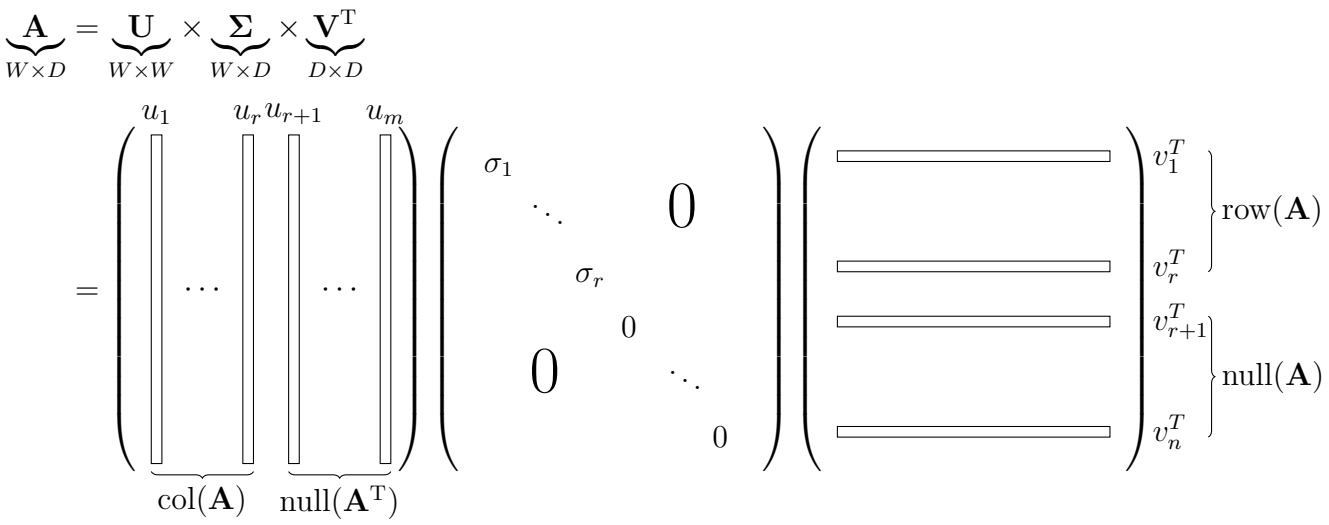
where $\mathbf{U}$ is a $W \times W$ orthogonal matrix whose columns are the eigenvectors of $\mathbf{A} \times \mathbf{A}^{\mathrm{T}}$. $\mathbf{V}$ is a $D \times D$ orthogonal matrix whose columns are the eigenvectors of $\mathbf{A}^{\mathrm{T}} \times \mathbf{A}$. $\mathbf{\Sigma}$ is a $W \times D$ diagonal matrix of the form:

$$\mathbf{\Sigma} = \begin{pmatrix} \sigma_1 & & & \\ & \ddots & 0 & \\ & & \ddots & \\ & 0 & & \sigma_r \\ & & & & 0 \end{pmatrix}$$

with $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r > 0$ and $r = \mathrm{rank}(\mathbf{A})$. $\sigma_1, \sigma_2, \cdots, \sigma_r$ are the square roots of the eigenvalues of $\mathbf{A}^{\mathrm{T}} \times \mathbf{A}$, which are termed as the singular values of $\mathbf{A}$.

[1]Some content has been borrowed from: http://www.cs.iastate.edu/~cs577/handouts/svd.pdf

The SVD factorization can be represented as follows:



Based on the matrix factorization scheme described above, we can deduct the following about the SVD matrix factorization algorithm. In the scheme above, null($\mathbf{A}$) represents the null space of the matrix $\mathbf{A}$. We can also see in the decomposition scheme above that the rank of the matrix $\mathbf{A}$ is equal to the rank of the diagonal matrix consisting of the singular values $\mathbf{\Sigma}$. This rank is equal to r. In the first matrix which is represented as $\mathbf{U}$, the column space of the $\mathbf{A}$ is spanned by the first r columns of the orthogonal matrix $\mathbf{U}$. The rest is the null space. The row space of $\mathbf{A}$ is spanned by the first r columns of the orthogonal matrix $\mathbf{V}$, and the null space of $\mathbf{A}$ is spanned by the last $n - r$ columns of $\mathbf{V}$.

3.3 Latent Class Analysis (LCA)

Latent Class Analysis (LCA) is a paradigm that relates a set of observed variables to the latent or hidden variables [99]. A latent variable is a discrete variable. The clusters or classes thus generated by the latent class analysis technique is called as a latent class.

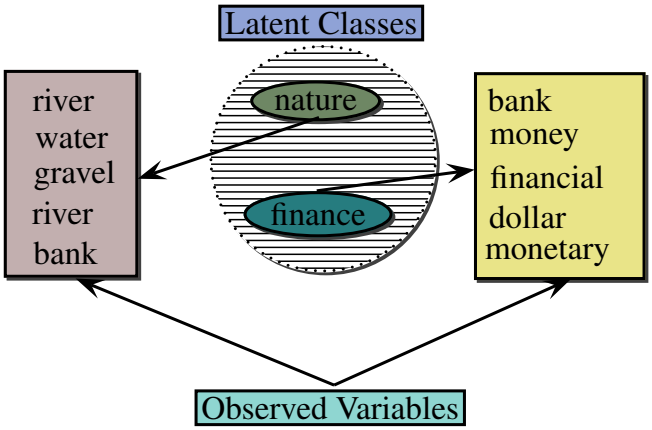


Figure 3.5: An example depicting the idea behind latent class analysis. Words such as "river" are observed variables, but "nature" and "finance" are latent variables because they are hidden, but words in each cluster together point out these two semantic names respectively.

The idea behind the LCA model is depicted in Figure 3.5, where the list of words generated in a cluster are observed words. But these lists of words belong to some latent class, for example, one list belongs to a latent class "finance" and the other belongs to "nature". Although these latent class names are not generated by the model, together they seem to point out to the subject/theme that has been manually listed for a cluster.

3.4 Latent Semantic Analysis (LSA)

The input to the LSA model is a term-document matrix $\mathbf{A}$. LSA uses the SVD algorithm (described in Section 3.2.1) along with a pre-defined number of latent factors, K[2], and reduces a high-dimensional vector space to a low-dimensional concept space. This new low-dimensional concept space brings about new term-term and term-document correlations which remains hidden in the original vector space. This results in the removal of noise too.

[2]The notation K used here as the total number of latent factors is different from K used in Chapter 4, which is the number of segment-topics.

LSA mainly performs the clustering of documents and terms in the low-dimensional latent concept space. It means that terms and documents which are "closely" or "semantically" related to each other cluster "close" to each other in the concept space. When the dimension of the space is small, then it helps get rid of the "curse-of-dimensionality", which is a problem especially in large datasets.

We show the LSA scheme in the matrix scheme below. First, we begin with a high-dimensional original vector space. The term-document matrix comprises of terms $v \in \{1, \cdots, W\}$ in the vocabulary in the rows, and the documents $d \in \{1, \cdots, D\}$ in the collection are represented in the columns.

$$\text{Terms} \overbrace{\begin{array}{c} \\ v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_W \end{array}}^{\begin{array}{ccccc} d_1 & d_2 & \cdot & \cdot & d_D \end{array}} \overbrace{\begin{pmatrix} 8 & 1 & 1 & 1 & 4 \\ 5 & 12 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}}^{\text{Documents}} \approx \mathbf{U}_k \times \mathbf{\Sigma}_k \times \mathbf{V}_k^{\mathrm{T}} \tag{3.2}$$

The term-document matrix is decomposed into three matrices as shown in Section 3.2.1, where now the matrix is approximated in a low-dimensional concept space with the number of latent concept factors pre-defined as $K = 3$ i.e. the dimension of the space has been pre-defined by the user. Also note that the discrete vector space will be transformed into a continuous low-dimensional latent concept space, and some values can be negative. The three matrices $\mathbf{U}_k$, $\mathbf{\Sigma}_k$ and $\mathbf{V}_k^{\mathrm{T}}$ can be expanded as:

$$\overbrace{\begin{array}{c} \\ \text{Terms} \end{array}}^{\text{Factors}} \left\{ \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{cccc} & k_1 & k_2 & k_3 \end{array} \\ \begin{array}{c} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_W \end{array} \begin{pmatrix} 1.00 & 0.91 & 1.00 \\ -0.44 & -0.57 & 0.84 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0.00 & 0.00 & 0.47 \end{pmatrix} \end{array} \right. \approx \mathbf{U}_k \tag{3.3}$$

The matrix comprising of the eigenvalues is written as:

$$\overbrace{\begin{array}{c} \\ \text{Factors} \end{array}}^{\text{Factors}} \left\{ \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{array}{c} k_1 \\ k_2 \\ k_3 \end{array} \begin{pmatrix} 3.34 & 0.00 & 0.00 \\ 0.00 & 2.54 & 0.00 \\ 0.00 & 0.00 & 1.001 \end{pmatrix} \end{array} \right. \approx \mathbf{\Sigma}_k \tag{3.4}$$

The final orthogonal matrix $\mathbf{V}^{\mathrm{T}}$ can be written as:

$$\overbrace{\begin{array}{c} \\ \text{Factors} \end{array}}^{\text{Documents}} \left\{ \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & \cdot & \cdot & d_D \end{array} \\ \begin{array}{c} k_1 \\ k_2 \\ k_3 \end{array} \begin{pmatrix} -0.19 & -0.05 & \cdot & \cdot & 0.10 \\ -0.01 & 0.43 & \cdot & \cdot & 0.52 \\ -0.03 & 0.45 & \cdot & \cdot & -0.64 \end{pmatrix} \end{array} \right. \approx \mathbf{V}_k^{\mathrm{T}} \tag{3.5}$$

3.4.1 Limitations of the LSA Model

The limitations of the LSA model are as follows:

- The model lacks a solid methodological foundation when it comes to matrices such as the term-document matrix, its application largely remains ad-hoc.
- The resulting low-dimensional vectors tend to lie in the negative side of the concept space, which does not generally make much sense in text processing.
- The model is unable to handle polysemy.

3.5 Probabilistic Latent Semantic Analysis (pLSA)

The pLSA model [101] attempted to solve some of the shortcomings in the LSA. The pLSA follows the paradigm of the latent class model. It is mainly an extension of the aspect model [104], [103]. The model can also be viewed as a matrix factorization method [62], [77], [61], which generates low-dimensional matrices which is an approximation of the original vector space. However, unlike SVD, the elements in the matrices are all positive just as in Non-negative Matrix Factorization (NMF) models [152], [41].

The pLSA model is characterized by the following generative process:

1. First select a document d in the corpus with probability $P(d)$
2. For each word w_i^d in the document d, irrespective of its position in the document
   (a) Select a latent variable z_i^d from a **Multinomial**$(P(z_i^d|d))$
   (b) Select a word w_i^d from a **Multinomial**$(P(w_i^d|z_i^d))$

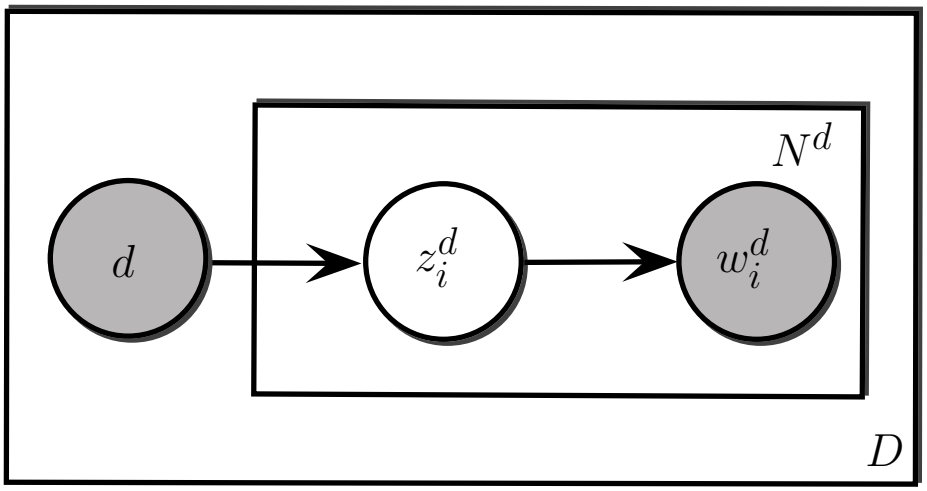


Figure 3.6: Graphical model in standard plate notation of the pLSA model. In a graphical model, plates signify repetition of data, and circles depict variables. The circles which are shaded in black are observed variables.

We have presented the graphical model of the pLSA model in standard plate notation in Figure 3.6. Mathematically, the pLSA model can be described as follows:

$$P(d, w_i^d) = P(d) \times P(w_i^d|d) \tag{3.6}$$

and, $P(w_i^d|d)$ can be expressed as:

$$P(w_i^d|d) = \sum_{z_i^d \in L} P(w_i^d, z_i^d, d) \tag{3.7}$$

The expression above can be equivalently written as:

$$\underbrace{P(w_i^d|d)}_{\text{Multinomial mixtures}} = \sum_{z_i^d \in L} \underbrace{P(w_i^d|d, z_i^d)}_{\text{Multinomials}} \times \underbrace{P(z_i^d|d)}_{\text{Mixing weights}} \tag{3.8}$$

Since the variable are conditionally independent (as shown in the graphical model), we can write:

$$P(w_i^d|d) = \sum_{z_i^d \in L} P(w_i^d|z_i^d) P(z_i^d|d) \tag{3.9}$$

The joint distribution can be written as:

$$P(w_i^d, d) = \sum_{z_i^d \in L} P(z_i^d) \times P(d|z_i^d) \times P(w_i^d|z_i^d) \tag{3.10}$$

The parameters of the model are $P(w_i^d|z_i^d)$ and $P(z_i^d|d)$. For $P(w_i^d|z_i^d)$, the number of parameters are $(W-1) \times L$, and for $P(z_i^d|d)$, the number of parameters are $D \times (L-1)$. The reason why we have $(W-1) \times L$ and $D \times (L-1)$ parameters is mainly because of the normalization constraint.

3.5.1 pLSA as a Matrix Factorization Model

pLSA can also be viewed as a matrix factorization technique, just like the SVD in LSA. The difference lies in the resultant vectors which are all positive in case of the pLSA model. In the scheme below, we present how pLSA can be viewed as a matrix factorization model where we obtain three matrices which brings out better correlations among the vectors in the low-dimensional space.

Assuming the number of factors as $K = 3$ in the pLSA model, which is pre-defined by the user. The original term-document matrix can be factorized as below:

$$\overbrace{\begin{array}{c} \\ \text{Terms} \end{array}}^{\text{Documents}} \left\{ \begin{array}{c} \\ \end{array} \right. \begin{array}{c} \\ \end{array} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & \cdot & \cdot & d_D \end{array} \\ \begin{pmatrix} 8 & 1 & 1 & 1 & 4 \\ 5 & 12 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{l} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_W \end{array} \approx P(\mathbf{w}|\mathbf{z}) \times P(\mathbf{z}) \times P(\mathbf{d}|\mathbf{z}) \tag{3.11}$$

The matrix comprising of the words' contribution in each latent class is written as below:

$$\overbrace{\text{Terms}\left\{\begin{array}{c} \\ v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_W \end{array}\right.}^{\text{Factors}} \begin{array}{c} \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{pmatrix} 1.00 & 0.91 & 1.00 \\ 0.44 & 0.57 & 0.84 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0.00 & 0.00 & 0.47 \end{pmatrix} \end{array} \approx P(\mathbf{w}|\mathbf{z}) \tag{3.12}$$

The matrix comprising of the eigenvalues can be written as:

$$\overbrace{\text{Factors}\left\{\begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \end{array}\right.}^{\text{Factors}} \begin{array}{c} \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{pmatrix} 0.34 & 0.00 & 0.00 \\ 0.00 & 0.54 & 0.00 \\ 0.00 & 0.00 & .001 \end{pmatrix} \end{array} \approx P(\mathbf{z}) \tag{3.13}$$

The final orthogonal matrix in this model can be written as:

$$\overbrace{\text{Factors}\left\{\begin{array}{c} \\ k_1 \\ k_2 \\ k_3 \end{array}\right.}^{\text{Documents}} \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & \cdot & \cdot & d_D \end{array} \\ \begin{pmatrix} 0.19 & 0.05 & \cdot & \cdot & 0.10 \\ 0.01 & 0.43 & \cdot & \cdot & 0.52 \\ 0.03 & 0.45 & \cdot & \cdot & 0.64 \end{pmatrix} \end{array} \approx P(\mathbf{d}|\mathbf{z}) \tag{3.14}$$

The pLSA model has some limitations such as the number of parameters grow linearly with the data. The model also is not a pure generative model for unseen documents, and the model tends to overfit.

3.6 Dirichlet Distribution

The probability density of a L dimensional Dirichlet distribution over the Multinomial distribution $P = (P_1, P_2, \cdots, P_L)$ is defined as:

$$\mathbf{Dir}(\alpha_1, \cdots, \alpha_L) = \frac{\Gamma(\sum_z \alpha_j)}{\prod_z \Gamma(\alpha_z)} \prod_{z=1}^{L} P_z^{\alpha_z - 1} \tag{3.15}$$

The parameters of the distribution shown above are defined by $\alpha_1, \cdots, \alpha_L$. The Dirichlet distribution is a conjugate prior of the Multinomial, so it can be regarded as a convenient choice as prior. It is so because this choice makes the mathematical derivation a lot easier. Each element in $\alpha_1, \cdots, \alpha_L$ is called a hyperparameter, and each hyperparameter α_z can be regarded as a form of a prior count for the number of times a topic z has been sampled in a document. For example, consider Equation 3.23, where $\alpha_{z_i^d}$ has been added as a prior count. The reason why a Dirichlet makes a convenient choice as a prior distribution is primarily because if the data data points come from a Multinomial distribution, and the prior distribution of those data points is considered as a Dirichlet, then the posterior distribution of the parameters is also a Dirichlet distribution. When all α_z are set to the same value, then the hyperprior is termed as symmetric.

$\alpha_1, \cdots, \alpha_L$ are also called as the concentration hyperparameters. If the concentration hyperparameter has a value less than 1, then the probability mass will be very concentrated, assuming that the distribution is discrete. If the value is greater than 1, then the mass will be more evenly distributed. We illustrate this phenomenon in

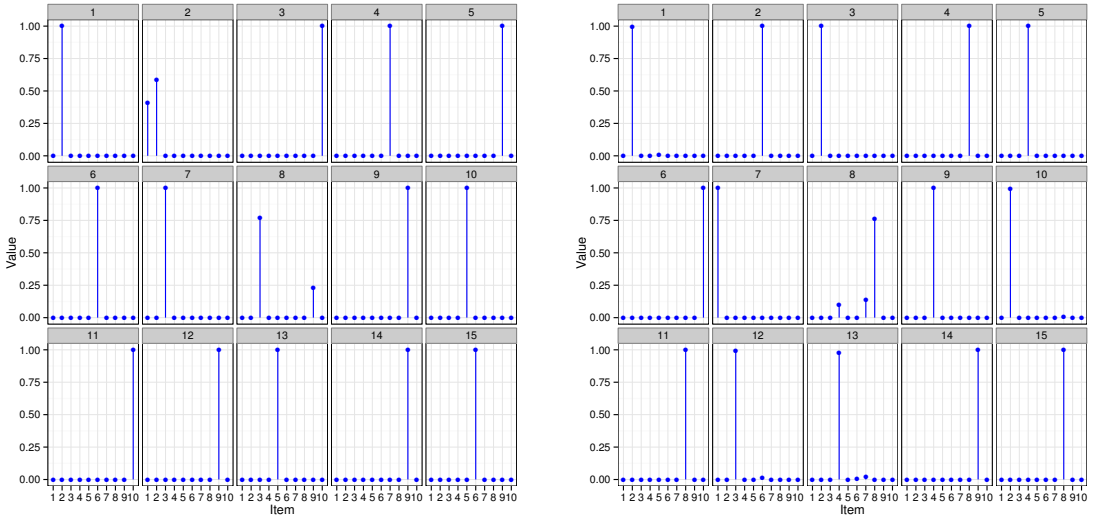


Figure 3.7: A plot showing 15 random draws from the Dirichlet distribution with $\alpha = 0.001$ on the left side and $\alpha = 0.01$ on the right. The dimension of the space is $L = 10$. We can see that in this plot the distribution tends to be very sparse with only few components with a high probability value. Others are almost close to 0.

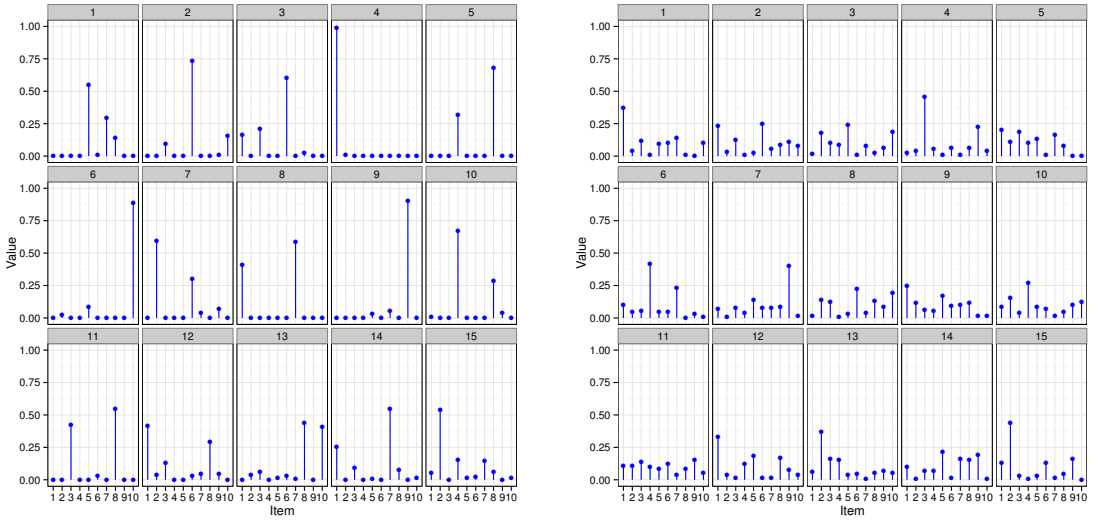


Figure 3.8: A plot showing 15 random draws from the Dirichlet distribution with $\alpha = 0.1$ on the left side and $\alpha = 1$ on the right. The dimension of the space is $L = 10$. We can see that in this plot the distribution tends to be very sparse with only few components with a high probability value. Others are almost close to 0.

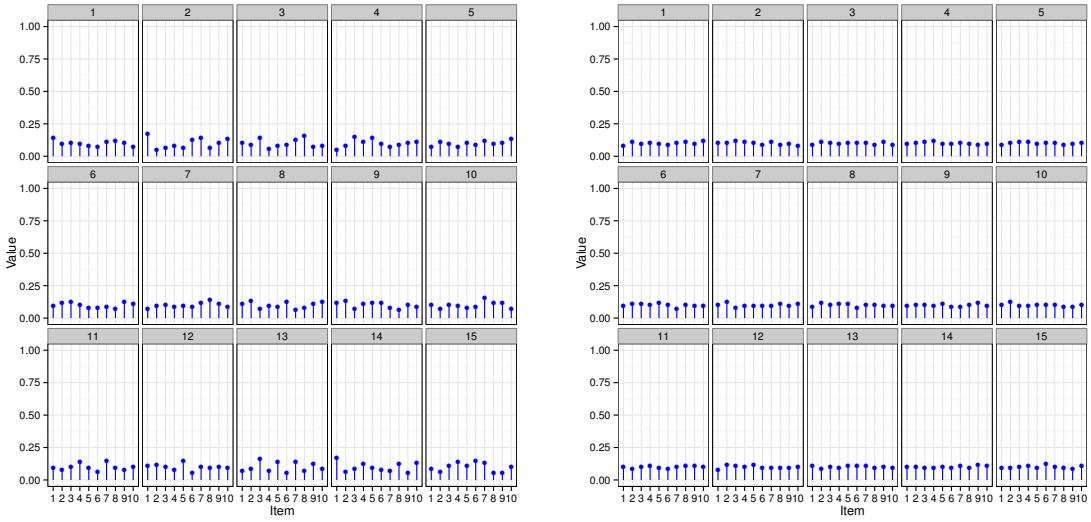


Figure 3.9: A plot showing 15 random draws from the Dirichlet distribution with $\alpha = 10$ on the left side and $\alpha = 100$ on the right. The dimension of the space is $L = 10$. We can see that in this plot the distribution tends to be very sparse with only few components with a high probability value. Others are almost close to 0.

Figure 3.7, 3.8 and 3.9. In all the three figures, we randomly draw 15 samples from a Dirichlet distribution with the dimension of the space equal to 10. In probabilistic topic modeling paradigm, this can be regarded as the number of topics pre-defined by the user. In Figure 3.6, when we set $\alpha = 0.001$, we notice that the distribution is very sparse with just one or two factors having a high value, while the rest are all close to zero. In probabilistic topic modeling paradigm, it could mean that if we choose the value of α very less, then the number of topics that describes each document will be less as documents will only describe about one or two topics. In case we choose a high value of α as those shown in Figure 3.9, the topics that each document will describe will be more. In practice, this does not make much sense as most documents generally describe about one or two topics, and rarely incoherently describe about several topics in one place.

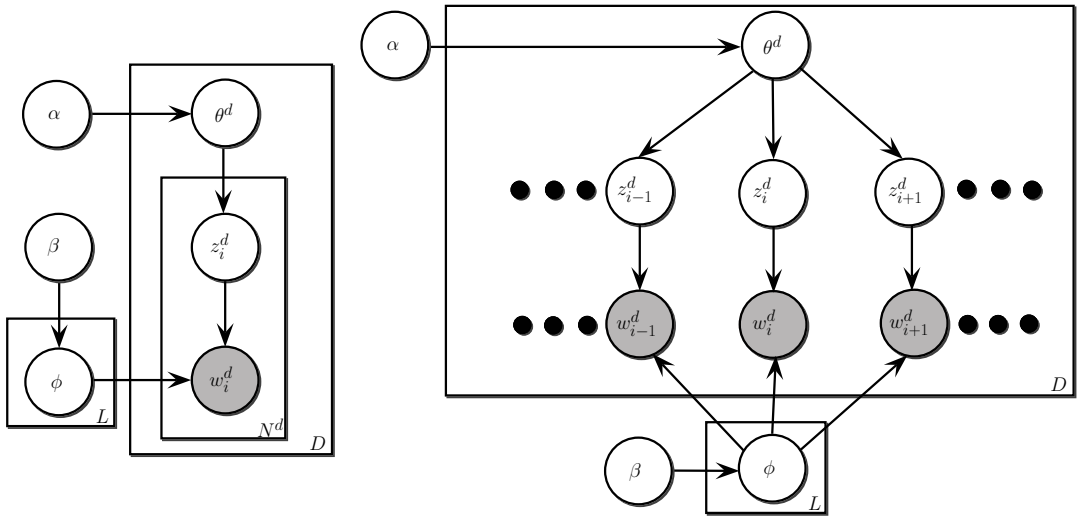


Figure 3.10: Latent Dirichlet Allocation model (LDA) in plate notation shown in the left, and its expanded notation shown on the right.

3.7 Probabilistic Unsupervised Topic Modeling

3.7.1 Latent Dirichlet Allocation Model

Topic models such as the LDA model are based upon the idea that documents exhibit multiple topics [237]. A topic, typically, is a probability distribution over words in the vocabulary. A topic model is a generative model for documents, it means that the model specifies a simple probabilistic procedure by which documents are generated. In order to build a new document from the seen past examples, one chooses a distribution over topics. Then, for each word in that document, one chooses a topic at random according to this distribution, and draws a word from that topic. Standard statistical techniques can be used to invert this process, inferring the set of topics that were responsible for generating a collection of documents. We present a simplest topic model, called LDA, which exemplifies the idea even further. The graphical model of the LDA model is presented in Figure 3.10, where we first present the model in standard plate notation. Then we expand the plate notation to show words in the document. In Figure 3.11, based on the generative process, we show

diagrammatically the LDA model.

The LDA model, which is shown in Figure 3.10, posits that documents exhibit multiple topics. The model describes the following generative process of each document in the corpus:

1. For each document d

   (a) Draw a topic proportion θ^d for a document d where $d \in [1, \ldots, D]$ from **Dirichlet** (α), where **Dirichlet** (α) is the Dirichlet distribution with parameter α on the per-document topic distributions,

   (b) Draw ϕ_k from **Dirichlet** (β) for each topic k, where ϕ_k is the word distribution for topic k and $k \in \{1, \ldots, L\}$. β is the parameter of the Dirichlet prior on the per-topic word distribution.

   (c) For each word w_i^d at position i in the document d

      i. Draw a topic z_i^d for each word w_i^d at position i in the document d from **Multinomial** (θ^d)

      ii. Draw a word w_i^d from **Multinomial** $(\phi_{z_i^d})$

When the parameters of the model are given, the joint distribution of a topic mixture θ^d, a set of topics $\boldsymbol{z}^d$ for that document d, and the words in that document $\boldsymbol{w}^d$ is given by:

$$P(\theta^d, \boldsymbol{z}^d, \boldsymbol{w}^d | \alpha, \beta) = P(\theta^d | \alpha) \prod_{n=1}^{N^d} P(z_i^d | \theta^d) P(w_i^d | z_i^d, \beta) \tag{3.16}$$

By taking the product of the marginal probabilities of all the documents in the

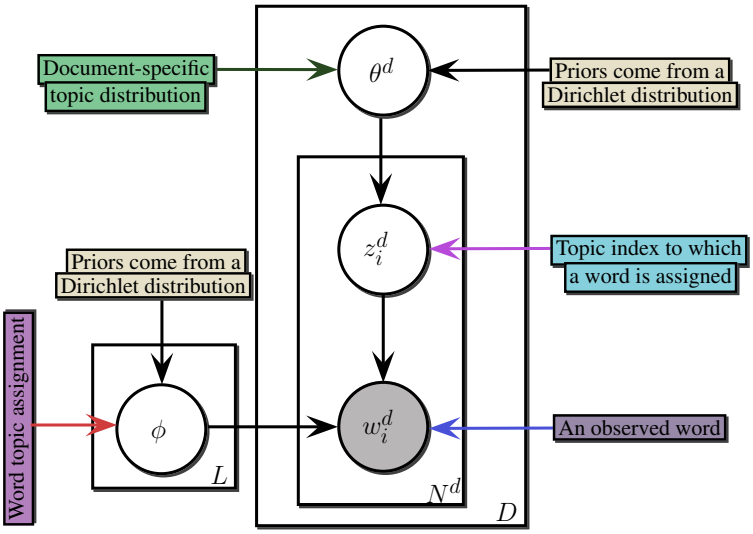


Figure 3.11: Explanation of the LDA graphical model.

collection S, the probability for the entire collection can be written as:

$$P(S|\alpha,\beta) = \prod_{d=1}^{D} \int P(\theta^d|\alpha) \left(\prod_{n=1}^{N^d} \sum_{z_i^d} P(z_i^d|\theta^d) P(w_i^d|z_i^d,\beta) \right) \mathrm{d}\theta^d \tag{3.17}$$

Computing the exact posterior distributions in the LDA model is intractable. Hence one has to resort to approximation techniques. Methods such as variational inference [23], [248], Gibbs sampling [86], [263], collapsed Gibbs sampling [208], [272] and many other techniques have been used in order to compute an approximation. The posterior distribution inferred by the LDA model is written as:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \alpha, \beta) = \frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta) P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\alpha, \beta)} \tag{3.18}$$

where $P_0(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}) = (\prod_{d=1}^{D} P(\theta^d|\alpha) \prod_{n}^{W} P(z_n^d|\theta^d)) \prod_{k=1}^{L} P(\phi_k|\beta)$ is the joint distribution defined in the model.

In Figure 3.12, we present the process of document generation. The figure shows four topics, where each topic describes about one thematically related content. For example, Topic 4 deals with some "data collection methodology using human anno-

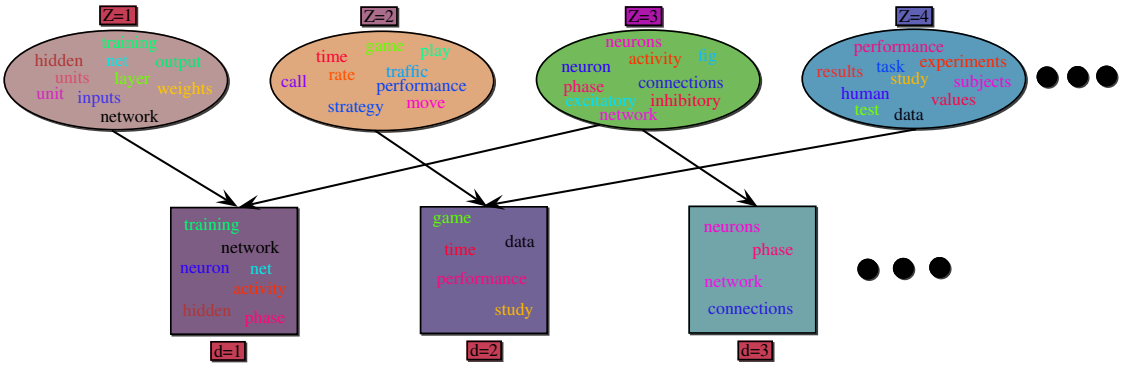


Figure 3.12: A figure showing the process of generation of documents from latent topics. In the figure we can see that documents are generated from words in topics. Some words in documents come from a mixture of topics, while some come from only one topic.

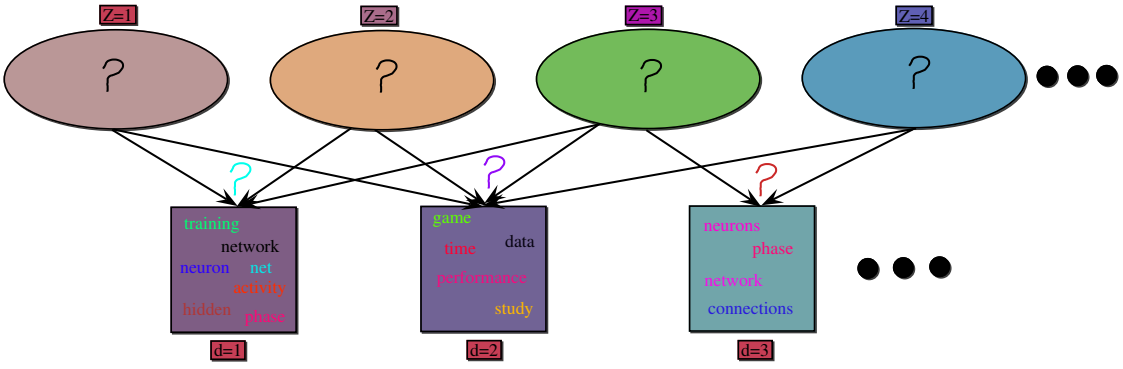


Figure 3.13: A figure showing the inference procedure in a topic model. During inference, we are given a set of documents with observed words. The task of the inference algorithm is to determine the probability of a word in a topic, the probability of a topic in a document, and also the topic assignment of a word in a document.

tators". Note that the words are represented as a bag-of-words where order of the words is not important. We use a colour coding scheme to let the reader know about which word comes from which topic. For instance, in Document 2, the word "game" from Topic 2, and "data" comes from Topic 4. These words are generated in the document according to certain probability value, for example, word "game" could be generated in the Document 2 with probability 0.001, and the word "data" could be generated in the document with probability 0.05. Therefore, documents can be generated from a topic depending upon the weight of the word from that topic. So if a document contains a mixture of several topics, then the document deals with multiple topics. This is the premise of the LDA model where it posits that a document exhibits multiple topics.

Figure 3.13 shows the process of statistical inference where we have the words in the documents which are observed variables. The task is then to infer the topic model which actually generated the data i.e. the process of model fitting. In this process, the probability distribution over words for each topic is computed, the distribution of topics over each document is realized, and the topic assignments of each word is generally formed. Such kind of inference procedures are mainly accomplished using some sampling techniques as computing the exact posterior distribution is intractable in case of probabilistic topic models.

In Figure 3.14, we present a topic visualization map that shows the words obtained in each topic and also how they are related to the other words in other topics. The notion is that if two topics are coherently related to each other, they will lie close to each other in the topic space. In this way, their Kullback-Leibler Divergence (KL-Divergence) value will be small in comparison to two topics which are semantically unrelated to each other.

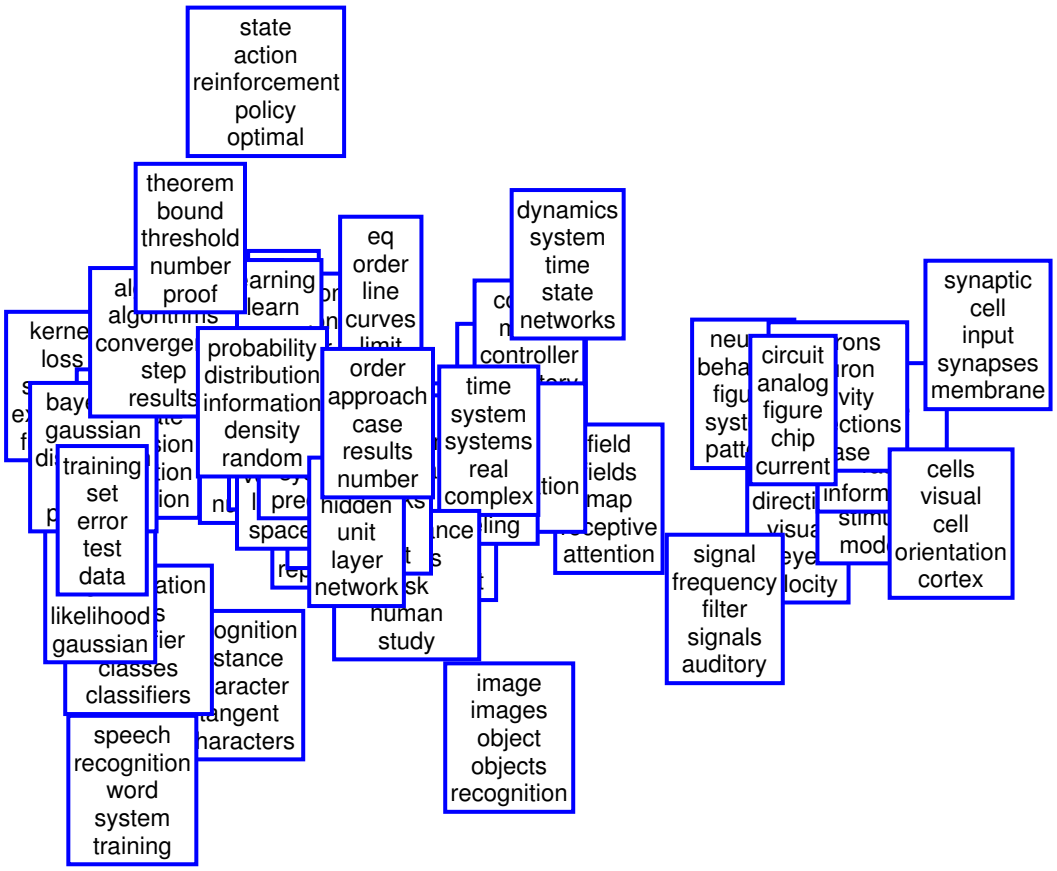


Figure 3.14: A two-dimensional map diagrammatically showing the topics obtained from the NIPS document collection when the number of topics K is set to 50. The idea behind the figure is that the latent topics which are semantically related to each other tend to cluster close to each other in the topic space, whereas substantially unrelated topical words tend to cluster far. This map has been created using the topic modeling toolbox[4] in MATLAB.

3.7.2 LDA as a Matrix Factorization Scheme

When the LDA model is viewed as a matrix factorization scheme, then the input to the model is the high-dimensional term-document matrix. Given a pre-defined number of latent topics, the model outputs two matrices, one is $\mathbf{\Theta}$, which consists of the topic distributions per document and the other is $\mathbf{\Phi}$, which consists of the word distributions per topic. In order to exemplify the idea further, we first show the input term-document matrix below:

$$\overbrace{\begin{array}{c} \\ \text{Terms} \end{array}}^{\text{Documents}} \left\{ \begin{array}{c} \\ \end{array} \right. \begin{array}{c} \begin{array}{ccccc} d_1 & d_2 & \cdot & \cdot & d_D \end{array} \\ \begin{pmatrix} 8 & 1 & 1 & 1 & 4 \\ 5 & 12 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \end{array} \begin{array}{l} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_W \end{array} \approx \mathbf{\Phi} \times \mathbf{\Theta} \tag{3.19}$$

$\mathbf{\Phi}$ can be represented as:

$$\overbrace{\begin{array}{c} \\ \text{Terms} \end{array}}^{\text{Factors}} \left\{ \begin{array}{c} \\ \end{array} \right. \begin{array}{c} \begin{array}{ccc} k_1 & k_2 & k_3 \end{array} \\ \begin{pmatrix} 1.00 & 0.91 & 1.00 \\ 0.44 & 0.57 & 0.84 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0.00 & 0.00 & 0.47 \end{pmatrix} \end{array} \begin{array}{l} v_1 \\ v_2 \\ \cdot \\ \cdot \\ \cdot \\ v_W \end{array} \approx P(\mathbf{w}|\mathbf{z}) \approx \mathbf{\Phi} \tag{3.20}$$

The matrix $\mathbf{\Theta}$ can be written as:

$$\text{Factors}\overbrace{\begin{Bmatrix} & \overbrace{\begin{matrix} d_1 & d_2 & \cdot & \cdot & d_D \end{matrix}}^{\text{Documents}} \\ \begin{matrix} k_1 \\ k_2 \\ k_3 \end{matrix} & \begin{pmatrix} 0.19 & 0.05 & \cdot & \cdot & 0.10 \\ 0.01 & 0.43 & \cdot & \cdot & 0.52 \\ 0.03 & 0.45 & \cdot & \cdot & 0.64 \end{pmatrix} \end{Bmatrix}}^{} \approx P(\mathbf{d}|\mathbf{z}) \approx \mathbf{\Theta} \tag{3.21}$$

Just as in the pLSA and NMF models, all the elements in the matrices are positive.

3.8 Topic Models with Word Order

3.8.1 Bigram Topic Model (BTM)

Wallach [252] proposed an extension to the LDA model called the BTM. The BTM model incorporated the Hierarchical Dirichlet Language Model (HDLM) [172] in the LDA model, and thus incorporated the notion of word order in the model. Instead of taking the input as the word-document co-occurrence matrix, the model takes the entire document as the input. The main idea behind the model is that a word is not only generated by the topic, but also generated by the previous word. This builds the notion of word dependence in sequence, and uses the First Order Markovian assumption.

The generative process of the model can be written as:

1. Draw **Multinomial** distributions (ϕ_{zw}) from a **Dirichlet** prior (β) for each topic and z_i^d and each word w_i^d.

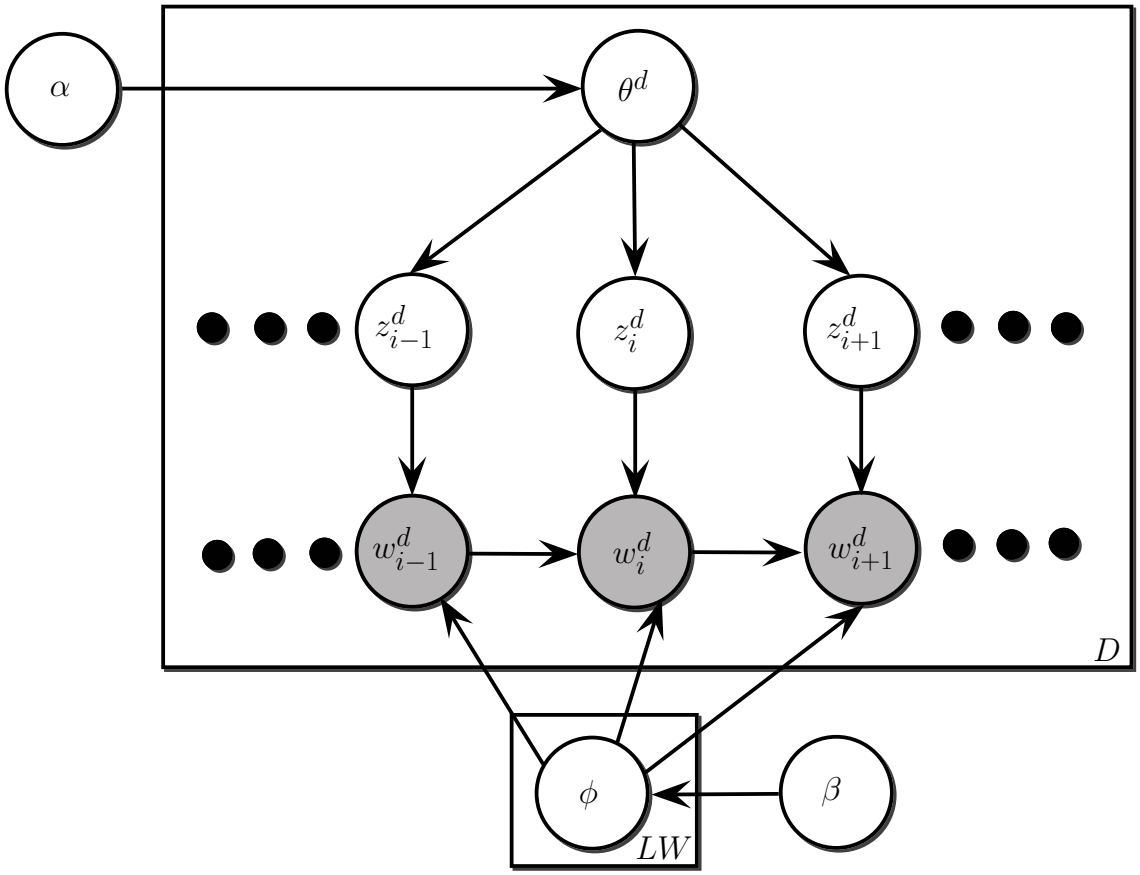


Figure 3.15: Graphical model showing the BTM.

2. For each document d,

   (a) Draw a **Multinomial** distribution (θ^d) from a **Dirichlet** prior (α).

   (b) For each word w^d_i in the document d,

      i. Draw z^d_i from **Multinomial** (θ^d)

      ii. Draw w^d_i from **Multinomial** ($\phi_{z^d_i w^d_{i-1}}$)

Computing the exact posterior distributions in the BTM model is intractable. Wallach resorted to EM [59] technique in order to approximate the posterior. A limitation of the EM algorithm is that it can get stuck in the local minima. For interested readers, we present the Gibbs sampling derivation for the BTM model in

Appendix C. Recently, Noji et al, [194] have shown better sampling schemes for the bigram topic model. The posterior distribution inferred by the BTM model is written as:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi} | \boldsymbol{W}, \boldsymbol{W}_{w_i}^{w_{i-1}}, \alpha, \beta) = \frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi} | \alpha, \beta) P(\boldsymbol{W}, \boldsymbol{W}_{w_i}^{w_{i-1}} | \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}, \boldsymbol{W}_{w_i}^{w_{i-1}} | \alpha, \beta)} \tag{3.22}$$

where $P_0(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}) = (\prod_{d=1}^{D} P(\boldsymbol{\theta}^d | \alpha) \prod_{n}^{W} P(z_i^d | \theta^d)) \prod_{k=1}^{K} P(\phi_k | \beta)$ is the joint distribution defined in the model.

3.8.2 LDA-Collocation Model (LDACOL)

The LDACOL model as described in [87], and shown in Figure 3.16 is a topic model that also relaxes the bag-of-words assumption and finds words as "single chunks", for example, "white house" rather than discover words independently. The model has a new set of random variables $\mathbf{x}$ in the LDA model. These random variables are binary variables which indicate the bigram status of a word i.e. whether word at position i denoted as w_i^d forms a bigram with the word at position $i-1$ denoted as w_{i-1}^d. If two words in sequence form a bigram then $x_i^d = 1$ else $x_i^d = 0$. In this way the model has an ability to generate both unigram and bigram words. Each word in the model has two assignments, one is the topic assignment and the other is the collocation assignment. If $x_i^d = 0$, then w_i^d is generated from a distribution that is dependent on w_{i-1}^d i.e. $P(w_i^d | w_{i-1}^d, x_i^d = 1)$ otherwise w_i^d is generated from a distribution of its topic, $P(w_i^d | z_i^d, x_i^d = 0)$. The value of x_i^d is chosen based on the previous word w_{i-1}^d, which is drawn from the distribution $P(w_i^d | w_{i-1}^d, x_i^d = 1)$. The LDACOL model is shown in Figure 3.16 in a standard plate notation where shaded variables denote observed variables and plates signify repetition, we can note that words in an order are connected with each other in sequence. Also, the bigram switch variables are

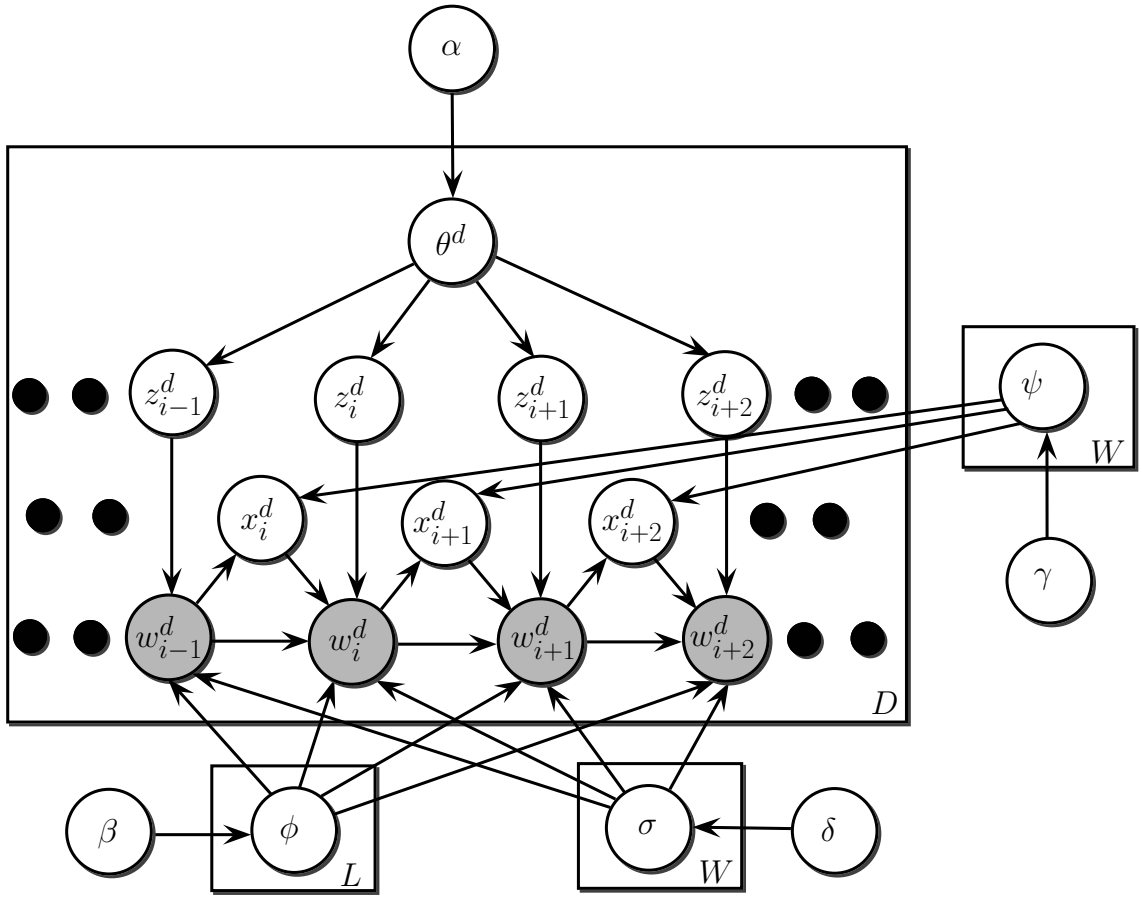


Figure 3.16: The graphical model in plate notation of the LDACOL model.

generated by the words as we move forward in the sequence. What we infer from the graphical model is that the LDACOL model is capable of capturing dependencies between the words in sequence. An advantage of the model is that it can generate both unigram and bigram words based on co-occurrences, and has empirically shown to perform better than the BTM model which only generates bigrams. BTM is also an n-gram topic model, but it does not possess the ability to generate variable gram words such as unigram or bigrams based on the local contextual information.

The generative process of the LDACOL model is defined as follows:

1. Draw **Multinomial** (ϕ_z) from **Dirichlet** (β) for each topic z

2. Draw **Bernoulli** (ψ_w) from **Beta** (γ) for each word w
3. Draw **Multinomial** (σ_w) from **Dirichlet** (δ) for each word w
4. For each document d
   (a) Draw **Multinomial** (θ^d) from **Dirichlet** (α)
   (b) For each word w_i^d in the document d
      i. Draw x_i^d from **Bernoulli** ($\psi_{w_{i-1}^d}$)
      ii. Draw z_i^d from **Multinomial** (θ^d)
      iii. Draw w_i^d from **Multinomial** ($\sigma_{w_{i-1}^d}$) if $x_i^d = 1$ else draw w_i^d from **Multinomial** ($\phi_{z_i^d}$)

As one can notice from the graphical model of the LDACOL model that every term in a bigram is not assigned to a topic. Only the first term has the topic assignment. Wang et al. [175] described a method to give the topic assignment to every term in the bigram in the LDACOL model, for example, they suggested that the topic of the first word in a bigram can be used to assign the topic of the other n-gram. But this assumption might not generate reasonable topical words. For longer phrases, which can be obtained by concatenating the consecutive bigram status variables (if consecutive $x_i^d = 1$), then in such phrase we can assume the highly occurring topic as the topic of the entire n-gram or phrase. Readers who are interested to have a look at the full derivation as requested to see Appendix D.

The Gibbs update formulae for the LDACOL model can be written as:

$$P(z_i^d|\mathbf{w}, \mathbf{z}_{\neg i}^d, \mathbf{x}, \alpha, \beta, \gamma, \delta) = \begin{cases} \underbrace{\left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{K} \left(\alpha_z + q_{dz}\right) - 1} \right) \times \frac{\beta_{w_i^d} + n_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^d v}) - 1}}_{\text{Same as in LDA}} & \text{if } x_i^d = 0 \\ \underbrace{\left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{K} \left(\alpha_z + q_{dz}\right) - 1} \right)}_{\text{Sample document-topic distribution}} \times \left(\frac{\delta_{w_i^d} + m_{w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W} (\delta_v + m_{w_{i-1}^d v}) - 1} \right) & \text{if } x_i^d = 1 \end{cases} \tag{3.23}$$

and,

$$P(x_i^d|\mathbf{w}, \mathbf{z}, \mathbf{x}_{\neg i}^d, \alpha, \beta, \gamma, \delta) = \underbrace{\left(\frac{\gamma_{x_i^d} + p_{w_{i-1}^d x_i^d} - 1}{\sum_{s=0}^{1} \left(\gamma_s + p_{w_{i-1}^d s}\right) - 1} \right)}_{\text{Sample bigram status variable}} \times \begin{cases} \frac{\beta_{w_i^d} + n_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^d v}) - 1} & \text{if } x_i^d = 0 \\ \frac{\delta_{w_i^d} + m_{w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W} (\delta_v + m_{w_{i-1}^d v}) - 1} & \text{if } x_i^d = 1 \end{cases} \tag{3.24}$$

And, the posterior estimates of $\theta, \phi, \psi, \sigma$ can be written as follows:

$$\hat{\theta}_z^d = \frac{\alpha_z + q_{dz}}{\sum_{t=1}^{K} (\alpha_t + q_{dt})} \tag{3.25}$$

$$\hat{\phi}_{zw} = \frac{\beta_w + n_{zw}}{\sum_{v=1}^{W} (\beta_v + n_{zv})} \tag{3.26}$$

$$\hat{\sigma}_{wv} = \frac{\delta_v + m_{wv}}{\sum_{v=1}^{W} (\delta_v + m_{wv})} \tag{3.27}$$

$$\hat{\psi}_{ws} = \frac{\gamma_k + p_{ws}}{\sum_{s=0}^{1} (\gamma_s + p_{ws})} \tag{3.28}$$

3.8.3 Topical N-gram Model (TNG)

The TNG model addresses some shortcomings in the LDACOL model. The TNG model gives the topic assignment to every term in a bigram, however the topic assignments may not be alike. The intuitive idea of the TNG model is that it is able to decide whether to form a bigram for the same two consecutive words depending on co-occurrences. The model also incorporates the bigram switching binary variables $\mathbf{x}$ which signify the bigram status of the word with the previous word in sequence. At the beginning of each document the model assumes a dummy word w_0^d. The model can be regarded as a more powerful generalization of the bigram topic model proposed in [252] and the LDACOL model. The model adopts a post-processing strategy to give the same topic assignment to every term in an n-gram, for example, in [261] the authors assumed the topic of the phrase as the topic of the "head noun". But the TNG model can be modified to give the same latent topic assignment to all the words in an n-gram. We present a modified version of the TNG model that gives the same topic assignment to all the words in an n-gram in Figure 3.18. In [166], the authors have suggested a way to solve some existing problems in the TNG model by incorporating the HPYP [203] in the n-gram topic model. We present the graphical model of the TNG model in Figure 3.17.

The generative process of the TNG model is described as:

1. Draw **Multinomial** (ϕ_z) from **Dirichlet** (β) for each topic z
2. Draw **Bernoulli** (ψ_{zw}) from **Beta** (γ) for each topic z and each word w
3. Draw **Multinomial** (σ_{zw}) from **Dirichlet** (δ) for each topic z and each word w
4. For each document d
   (a) Draw **Discrete** (θ^d) from **Dirichlet** (α)

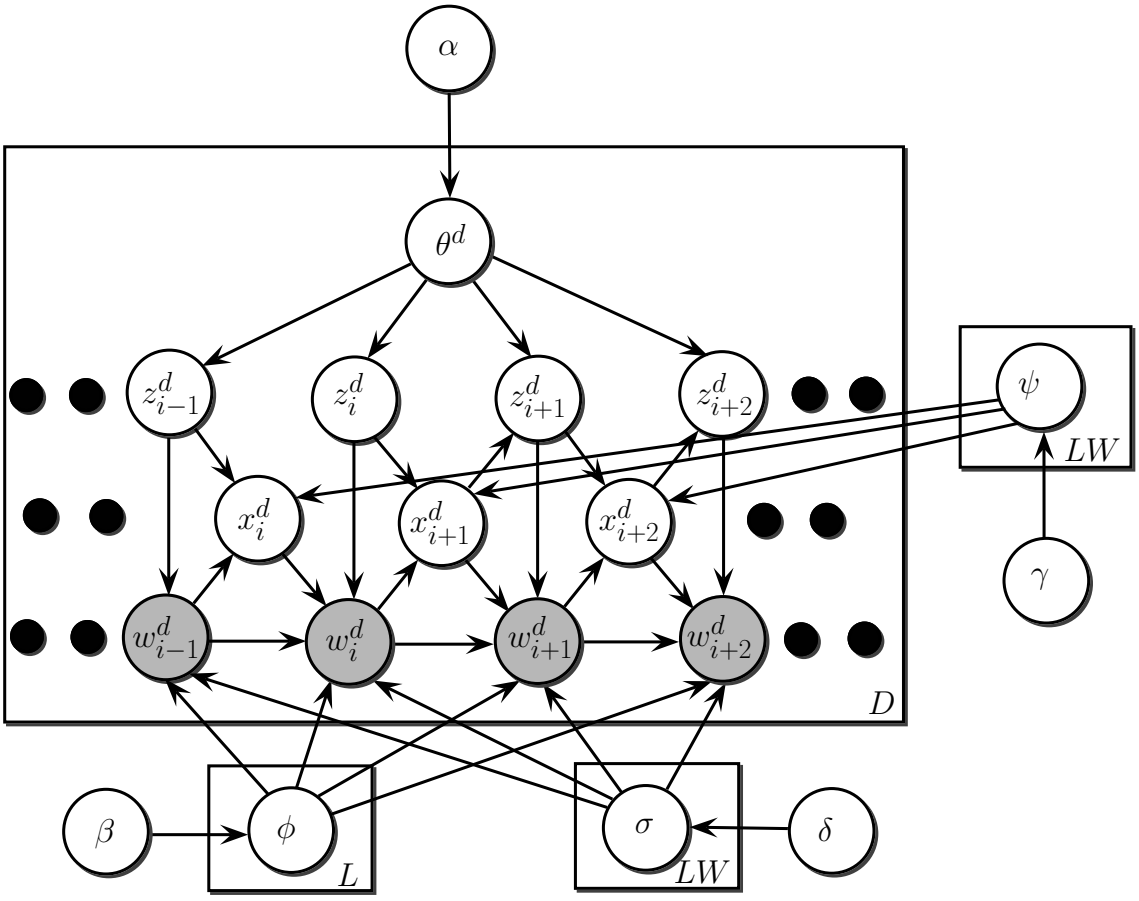


Figure 3.17: The graphical model in plate notation of the `TNG` model.

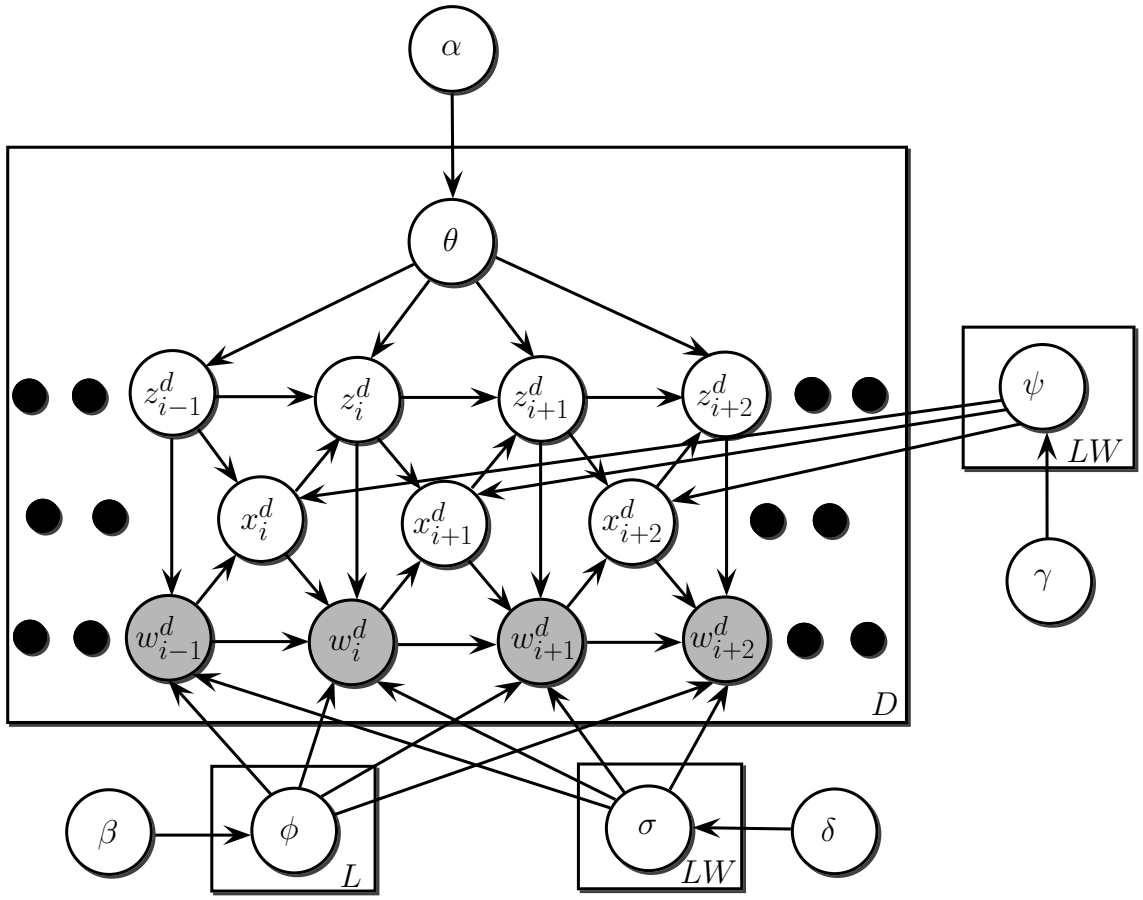


Figure 3.18: A graphical model showing a modified `TNG` model. This model, in contrast, to the previous graphical model has the ability to give the same topic assignment to the words in a bigram.

(b) For each word w_i^d in document d

i. Draw z_i^d from **Multinomial** (θ^d)

ii. Draw x_i^d from **Bernoulli** $(\psi_{z_{i-1}^d w_{i-1}^d})$

iii. Draw w_i^d from **Multinomial** $(\sigma_{z_i^d w_{i-1}^d})$ if $x_i^d = 1$ else draw w_i^d from **Multinomial** $(\phi_{z_i^d})$

The Gibbs update equations can be written as:

$$P(z_i^d|\mathbf{w}, \mathbf{z}_{\neg i}^d, \mathbf{x}, \alpha, \beta, \gamma, \delta) = \left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{K} \left(\alpha_z + q_{dz}\right) - 1} \right) \times \begin{cases} \frac{\beta_{w_i^d} + n_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^d v}) - 1} & \text{if } x_i^d = 0 \\ \frac{\delta_{w_i^d} + m_{z_i^d w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W} (\delta_v + m_{z_i^d w_{i-1}^d v}) - 1} & \text{if } x_i^d = 1 \end{cases} \tag{3.29}$$

and,

$$P(x_i^d|\mathbf{w}, \mathbf{z}, \mathbf{x}_{\neg i}^d, \alpha, \beta, \gamma, \delta) = \left(\frac{\gamma_{x_i^d} + p_{z_{i-1}^d w_{i-1}^d x_i^d} - 1}{\sum_{s=0}^{1} \left(\gamma_s + p_{z_{i-1}^d w_{i-1}^d s}\right) - 1} \right) \times \begin{cases} \frac{\beta_{w_i^d} + n_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W} (\beta_v + n_{z_i^d v}) - 1} & \text{if } x_i^d = 0 \\ \frac{\delta_{w_i^d} + m_{z_i^d w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W} (\delta_v + m_{z_i^d w_{i-1}^d v}) - 1} & \text{if } x_i^d = 1 \end{cases} \tag{3.30}$$

The posterior estimates of $\theta, \phi, \psi, \sigma$ can be written as follows:

$$\hat{\theta}_z^d = \frac{\alpha_z + q_{dz}}{\sum_{t=1}^{K} (\alpha_t + q_{dt})} \tag{3.31}$$

$$\hat{\phi}_{zw} = \frac{\beta_w + n_{zw}}{\sum_{v=1}^{W} (\beta_v + n_{zv})} \tag{3.32}$$

$$\hat{\sigma}_{zwv} = \frac{\delta_v + m_{zwv}}{\sum_{v=1}^{W}(\delta_v + m_{zwv})} \qquad (3.33) \qquad \hat{\psi}_{zws} = \frac{\gamma_k + p_{zws}}{\sum_{s=0}^{1}(\gamma_s + p_{zws})} \qquad (3.34)$$

3.9 Topic Segmentation Models

We describe an existing topic segmentation model known as LDSEG [230], shown in Figure 3.19, which assumes that the order of words in a segment is not important. LDSEG model detects the boundaries of text where topics change. The text within the respective boundaries is called a segment. Each segment is assigned to a super-topic from a predefined number of super-topics. Then, each segment is modeled based on its word content. The latent topics in each segment are called word-topics. Each super-topic is assumed to be a mixture of word-topics where the mixture coefficients uniquely specify the super-topic. Unigrams are assigned to word-topics. In the graphical model, $\mathbf{z}$ is the word-topic variable for the corpus. $\mathbf{y}$ is the super-topic variable for the corpus. S denotes the number of segments. N_s^d is the number of words in each segment. K is the number of super-topics. This model assumes "sentences" as a segment unit. Hence the super-topics will be assigned to sentences. LDSEG orders sentences of each document and assumes a Markov structure on the topic distribution of sentences. There is a binary switching variable $\mathbf{c}$ for the topic of each sentence. The word probabilities are modeled conditioned on the topics with a $L \times W$ matrix. The model assumes a Dirichlet prior for drawing the parameter of word distribution. τ defines the mixing proportion of the super-topics in document. θ is the mixing proportion of the word-topics in the text segment. π defines the parameter of the Binomial distribution. ρ constitutes the parameter of the Dirichlet prior on the super-topics. Ω is defined as the prior of the Binomial distribution. α is a matrix of $K \times L$ dimensions, row i represents the mixing proportion of the word-topics in the super-topic i. The Dirichlet prior parameter α is estimated during the learning phase.

β is the parameters of the prior probability for distribution of words conditioned on the word-topics. This variable specifies the mixing proportion of the word-topics in the segment s. The mixing proportion depends on the super-topic from which the current segment of text has been generated from. Apart from segmentation, the model can also cluster documents based on the super-topics. Approximate inference in the model is achieved through moment matching algorithm.

In Figure 3.20, we also present the component of the model that generates word-topics. We can see from the component that the portion matches the LDA model, and hence it generates unigrams in each segment.

In Figure 3.21, the component that generates the super-topics is highlighted. The variable **c** gives the segment change-points in the document, and the ordering of this variable in the document tells us where the segment changes in the document.

The generative process of the model can be written as:

1. Draw τ from **Dirichlet**(ρ), where τ is the mixing proportion of the segment-topics in a document, and ρ is the parameter of the Dirichlet prior on the super-topics .

2. Draw **Discrete**(ϕ_z) from **Dirichlet**(β) for each word-topic z, where ϕ is the parameter of the Multinomial distribution of word conditioned on the word-topics. β is the parameter of the prior probability for distribution of the words conditioned on the word-topics.

3. For each segment s from S

   (a) Draw y_s (i.e. the same super-topic) for s as its previous super-topic y_{s-1} with probability $P(c_s = 1) = \pi$, where π is the parameter of the Binomial distribution.

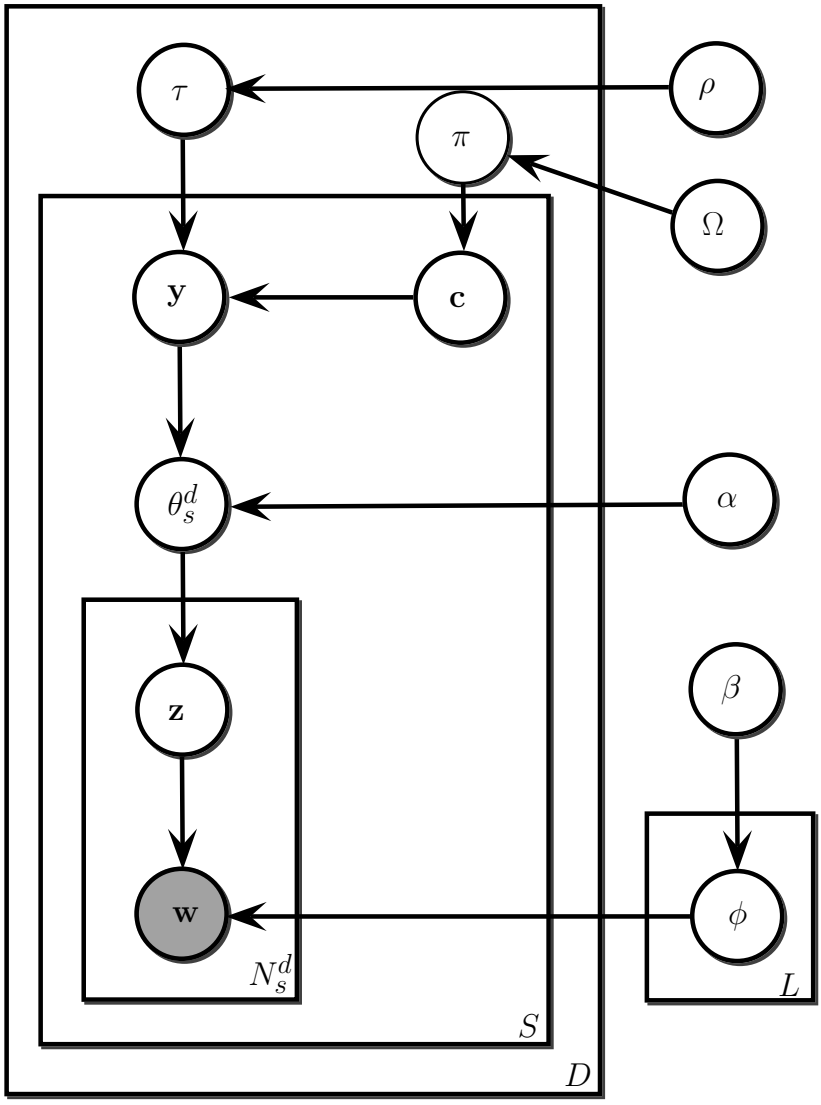


Figure 3.19: The graphical model of the LDSEG model in plate notation. The model depicts a hierarchy of two topic levels, one of which generates the super-topic and the other generates the word-topic. The variable c is the segment change-point variable.

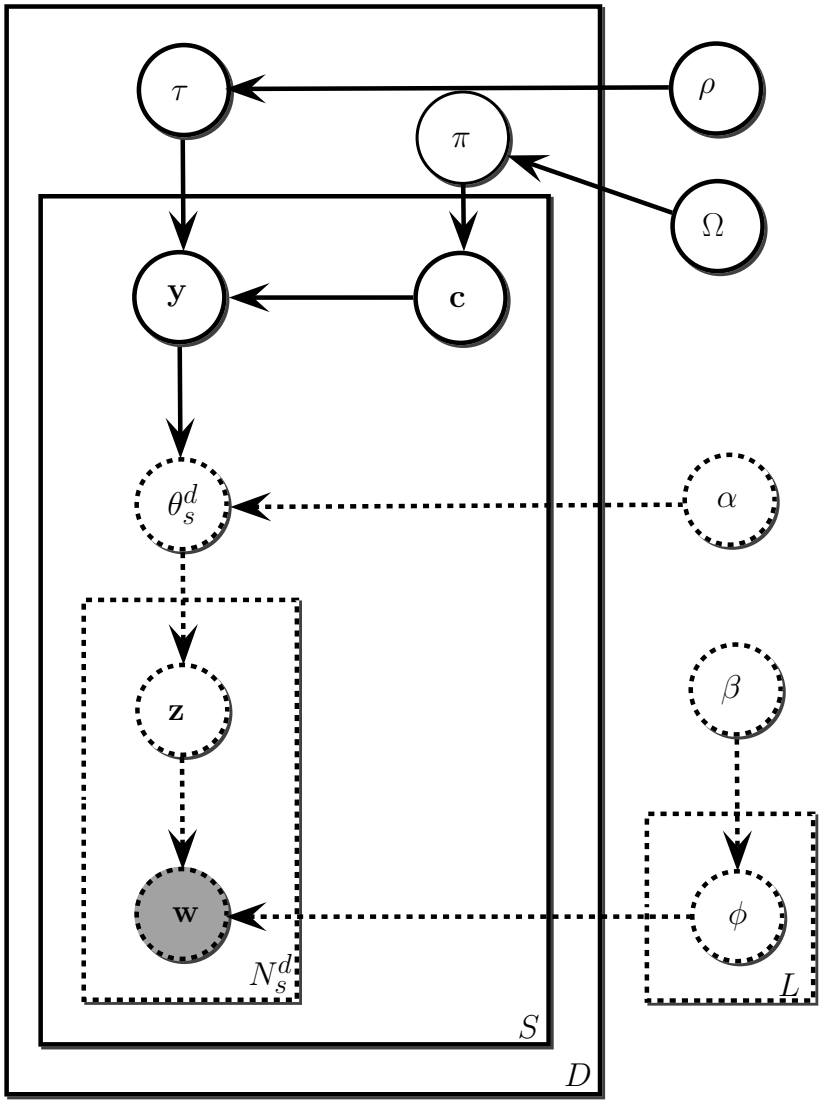


Figure 3.20: The graphical model of the `LDSEG` model where the component that generates the super-topics is highlighted.

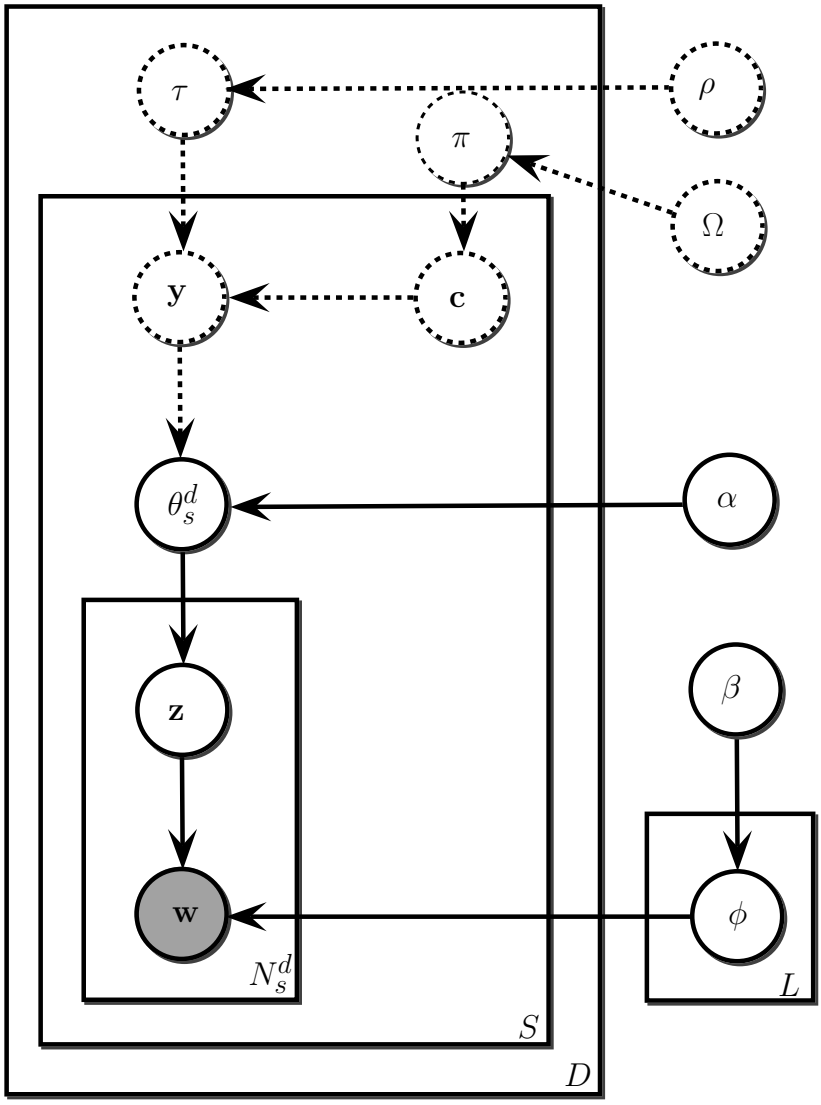


Figure 3.21: The graphical model of the `LDSEG` model where the component that generates the word-topics is highlighted.

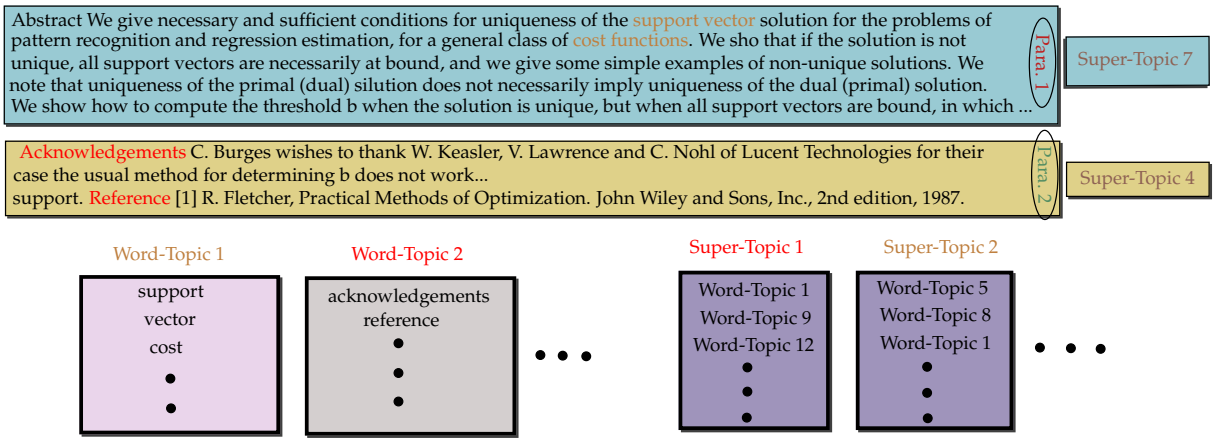


Figure 3.22: An illustration as to how the LDSEG model segments a document into two paragraphs with different super-topics. The super-topics in turn consist of the mixture of word-topics, and word-topics contain unigram words.

(b) Otherwise, draw a segment-topic for the segment y_s from **Multinomial**(τ)

(c) Draw $\theta^{(s)}$ from **Dirichlet**(α, y_s), where α is a $K \times L$ matrix where each row represents the mixing proportion of the word-topics in a super-topic.

(d) For each N_s^d words in the segment s

i. Draw z_{si}^d from **Discrete**$(\theta^{(s)})$, where $\theta^{(s)}$ is the mixing proportion of the word-topics in the text segment s. z_{si}^d is the word-topic assignment for the word w_{si}^d in segment s of document d.

ii. Draw w_{si}^d from **Discrete**$(\phi_{z_{si}^d})$

In Figure 3.22, we present an illustration about the hierarchy of topics generated by the LDSEG model. In this illustration, we have considered a segment as a paragraph. We see from the figure that two paragraphs in sequence in a document belong to different super-topics. It means that there is a change in the super-topic from one segment to another in sequence. Then we also notice that each segment-topic comprises of a mixture of word topic, and word-topics consist of unigram words.

3.10 Bayesian Nonparametrics in Topic Modeling

Finding the adequate complexity of a real dataset using a parametric model becomes very difficult because such models assume a restricted functional form. Some have suggested to use cross-validation scheme [298] in order to find the number of topics which may represent the true underlying characteristic of the data. However, such cross-validation schemes might be too time-consuming and computationally demanding [56], especially when dealing with a large document collection. In contrast, nonparametric models do not assume such restriction, and thus allow the model complexity to grow with the data. However, it should be noted that nonparametric models are not free of parameters. One needs parameters for computationally tractable representation. But as stated earlier, the number of parameters in a nonparametric model is not bounded, and grows with data characteristics. Nonparametric Bayesian models typically learn distributions on function spaces and can thus involve infinitely many parameters.

We will begin with the description of the related models, and subsequently use them as a basis to describe our model which is a nonparametric extension of the existing Bayesian nonparametric topic models.

3.10.1 Dirichlet Processes (DP)

A DP [68] is a distribution over distributions where a draw from this distribution is also a distribution. Dirichlet processes are often used in Bayesian nonparametrics. The Dirichlet process comprises of Dirichlet distributed finite dimensional marginal distributions. Draws from a Dirichlet process are discrete. They are not pre-defined by a fixed number of parameters.

Let G be a random distribution which is distributed according to a DP. Let H

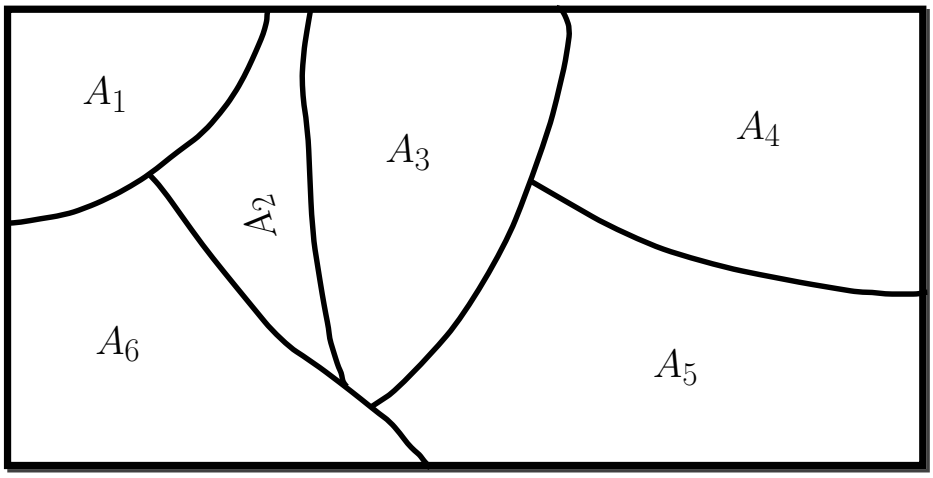


Figure 3.23: An example of partitions induced by a Dirichlet process.

be a distribution over Θ and α be a positive real number. Suppose there is a finite measurable partition $A_1, A_2, \cdots, A_r$, as shown in Figure 3.23, of Θ and the vector $(G(A_1), G(A_2), \cdots, G(A_r))$ is also random since G is random. This is written as $G \sim \text{DP}(\alpha, H)$ with a base distribution H, a concentration parameter α. G is Dirichlet process distributed if $G(A_1), G(A_2), \cdots, G(A_r) \sim \text{Dir}(\alpha H(A_1), \alpha H(A_2), \cdots, \alpha H(A_r))$ for every finite measurable partition $A_1, A_2, \cdots, A_r$ of Θ.

The base distribution H is the mean of the DP. The concentration parameter can be regarded as the inverse variance. Inquisitive readers are requested to consult [245] for more technical details. The stick-breaking representation [229], the Polyà Urn model [108], and the Chinese Restaurant Process (CRP) [3] metaphors can be used to describe the Dirichlet Process.

The following illustration will exemplify the concept of the DP even further. This will help us understand some of the sophisticated techniques later in this thesis. The codes for plots have been used from here[5] in order to explain the concept of the DP.

Consider a Polyà Urn model which is run several times for the same base distribution H. In the plots shown below, the base distribution is considered as a unit

[5]http://blog.echen.me/2012/03/20/infinite-mixture-models-with-nonparametric-bayes-and-the-dirichlet-process/

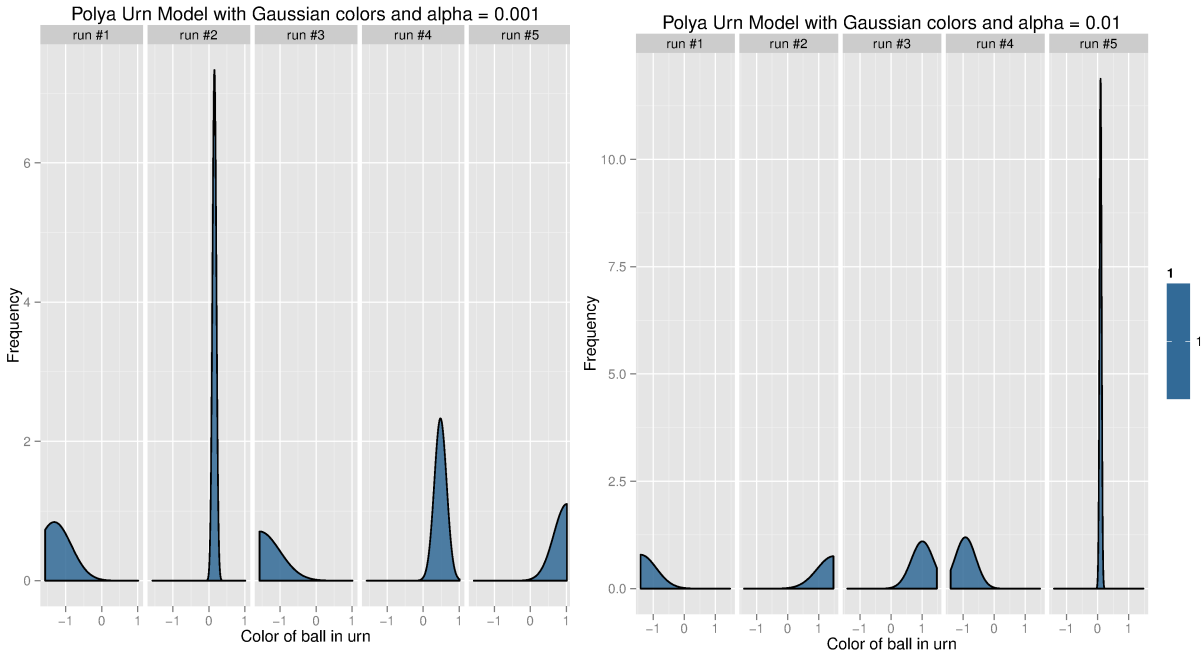


Figure 3.24: The figure depicts the density plots by sampling from the Polyà Urn model where the base distribution is considered as a standard unit normal. The concentration parameter α is varied to show the effect. If the concentration parameter is small, then more and more points tend to cluster in the same cluster and this results in less clusters, but if the concentration parameter is large then points are almost evenly distributed across the clusters. In the figure above, when $\alpha = 0.001$, we can see that points are concentrated only in a few clusters. Then we increase $\alpha = 0.01$ and points tend to disperse.

normal distribution. In each run (out of total five runs), there will be a different distribution of colours in the urn. This is because the entire process is random. Consider the following probability density plots generated by sampling from the Polyà Urn model with the standard unit normal as the base distribution. The number of coloured balls chosen in the simulation is 100.

Figures 3.24, 3.25, 3.26 depict some interesting patterns that is obtained by varying the concentration parameter. We see that as alpha increases which means that we are sampling more new coloured balls from our base distribution, the colors in the urn tend to a unit normal, which is just as our base distribution.

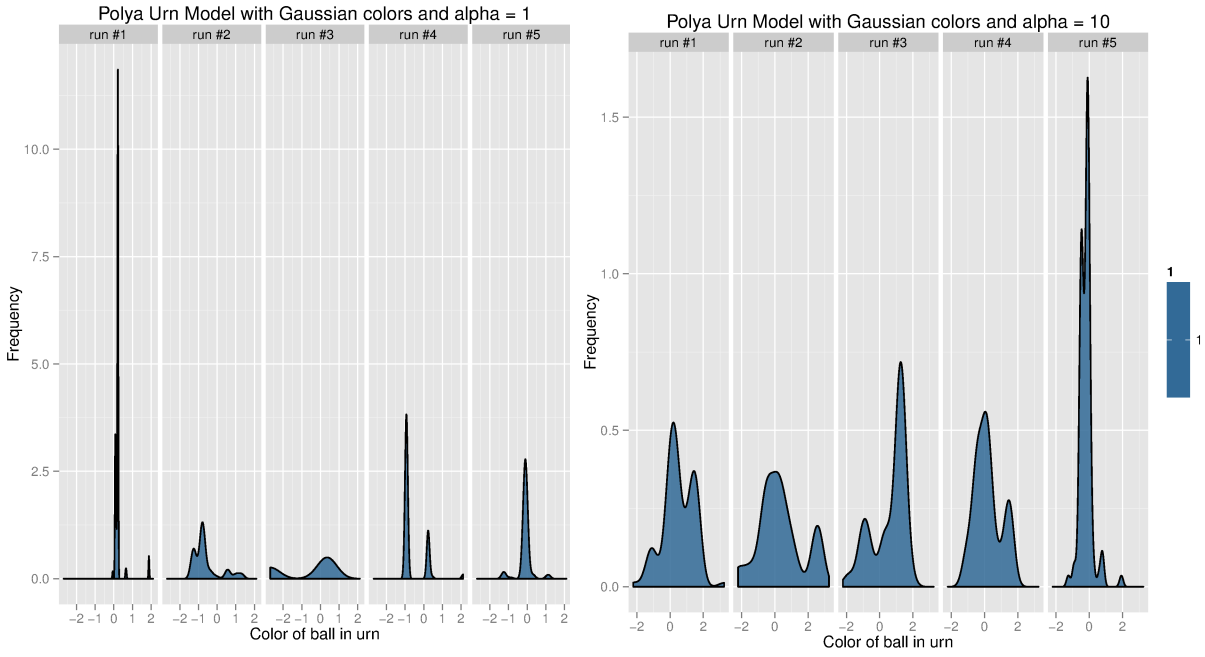


Figure 3.25: The figure depicts the density plots by sampling from the Polyà Urn model where the base distribution is considered as a standard unit normal. The concentration parameter α is varied to show the effect. If the concentration parameter is small, then more and more points tend to cluster in the same cluster and this results in less clusters, but if the concentration parameter is large then points are almost evenly distributed across the clusters. In the figure above, we can see that as compared to $\alpha = 10$, when $\alpha = 1$, the points are less dispersed.

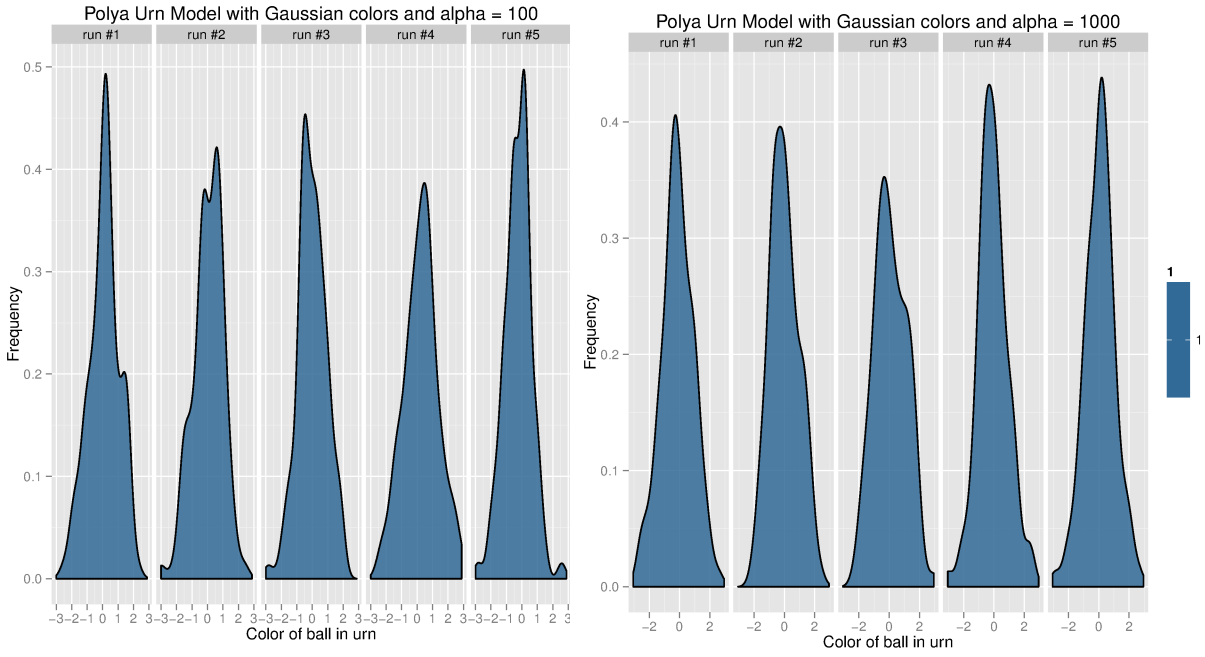


Figure 3.26: The figure depicts the density plots by sampling from the Polyà Urn model where the base distribution is considered as a standard unit normal. The concentration parameter α is varied to show the effect. If the concentration parameter is small, then more and more points tend to cluster in the same cluster and this results in less clusters, but if the concentration parameter is large then points are almost evenly distributed across the clusters. In the figures above, we can see that in both the cases the points are almost equally dispersed across the urns.

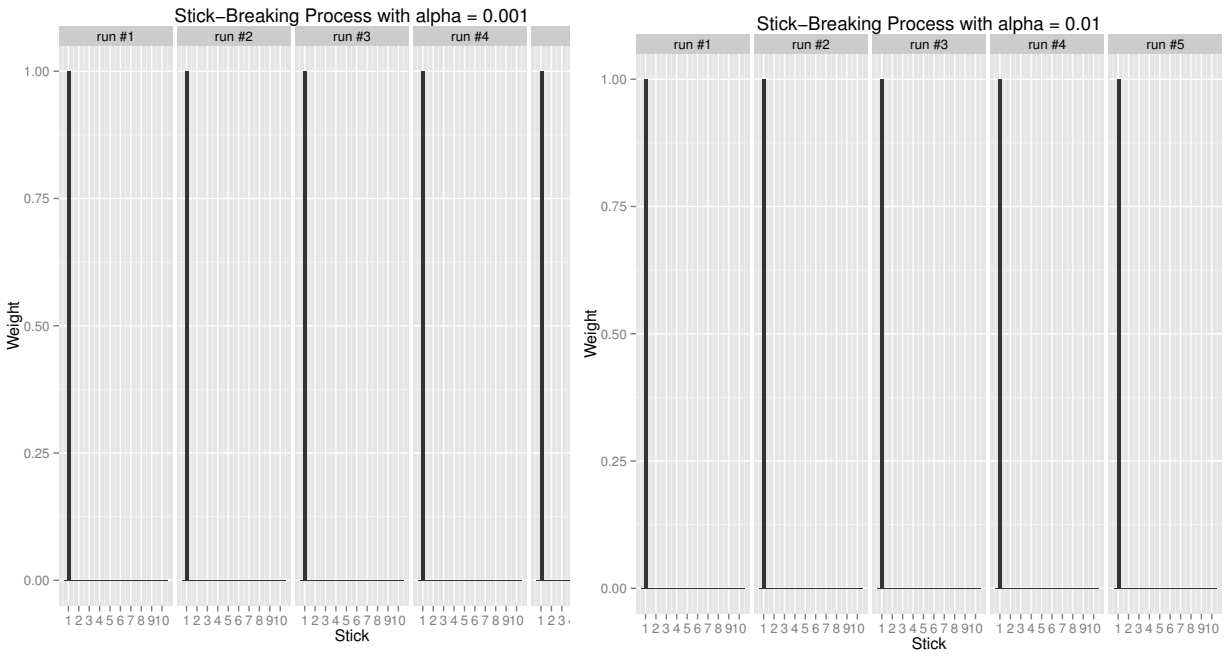


Figure 3.27: The figure depicts the stick breaking phenomenon where a unit length stick is repeatedly broken into smaller pieces. The concentration parameter controls the weights of the broken sticks, for example, in the figure above where the value of α is low, the stick weights are concentrated on the first few weights which suggests that the data points are concentrated on a few clusters.

3.10.2 Stick Breaking Construction

The stick breaking metaphor can be described in the following way. Consider a stick of unit length. In order to determine from where the stick has to be broken, we generate a random number from a Beta distribution which gives the point from where the stick has to be broken. Then the remaining stick is our stick which we will break in our next sampling step. If we iteratively keep on generating samples from the Beta distribution and we keep on breaking the sticks based on the numbers thus generated, in the end, we can obtain the proportion of the points that should belong to certain group.

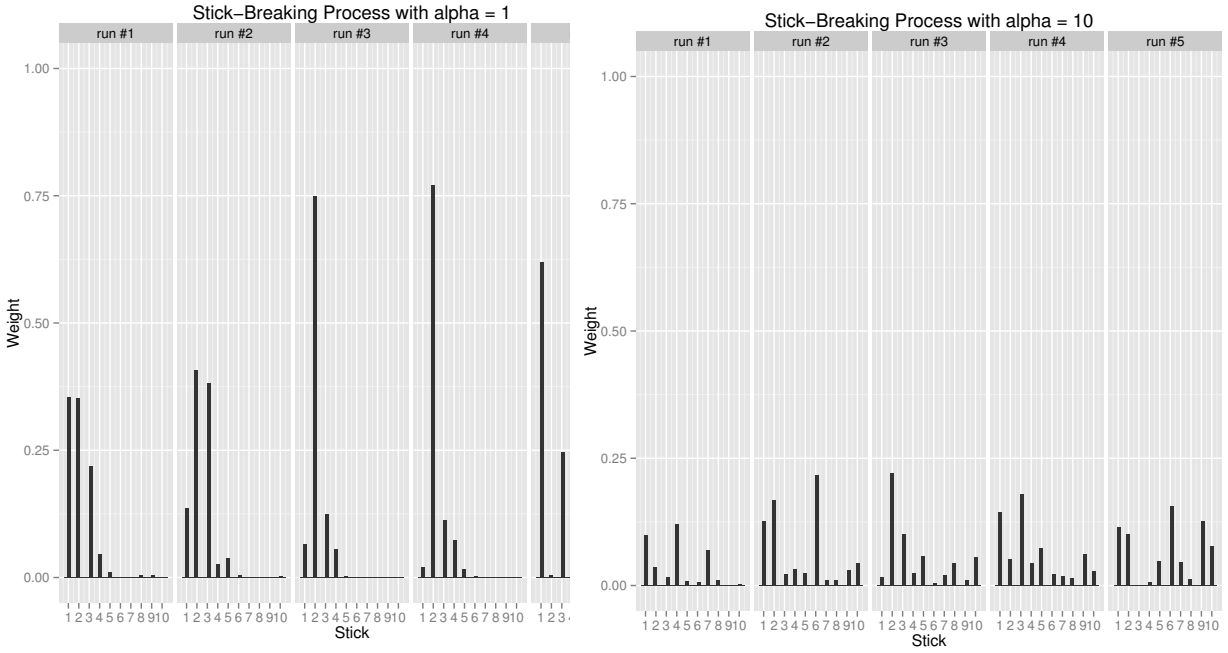


Figure 3.28: The figure depicts the stick breaking phenomenon where a unit length stick is repeatedly broken into smaller pieces. The concentration parameter controls the weights of the broken sticks.

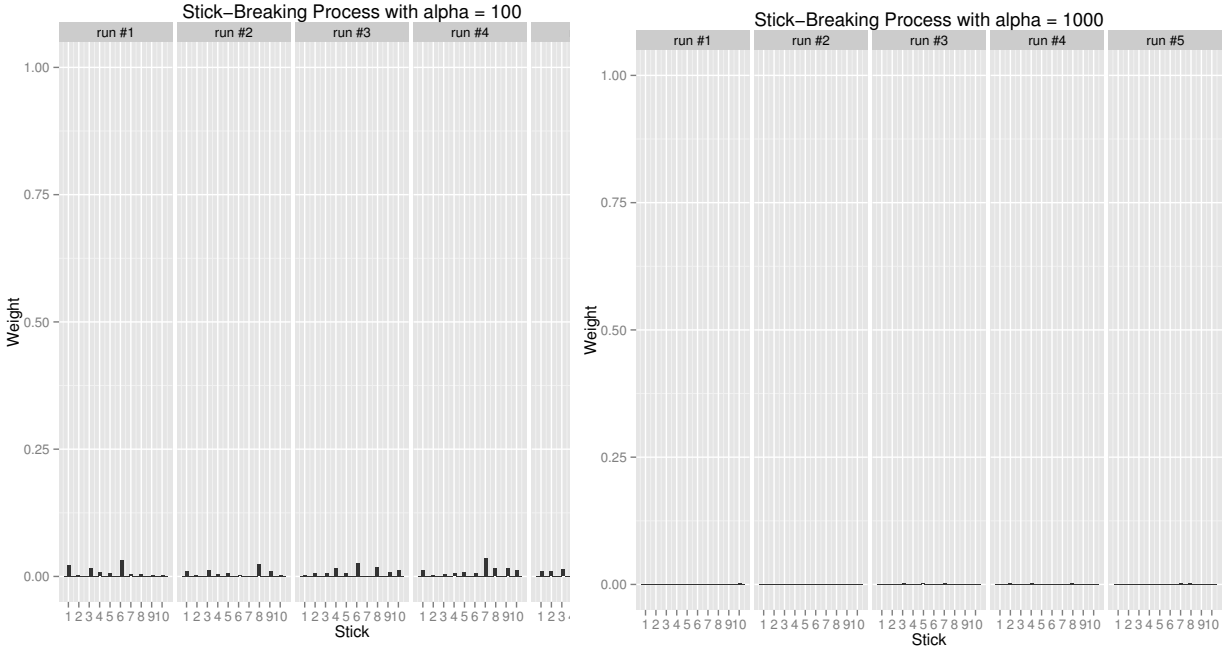


Figure 3.29: The figure depicts the stick breaking phenomenon where a unit length stick is repeatedly broken into smaller pieces. The concentration parameter controls the weights of the broken sticks, for example, in the figure above where the value of α is high, the stick weights are dispersed evenly.

3.10.3 The Dirichlet Process Mixture Model (DPMM)

In Dirichlet process mixture model, the Dirichlet process (DP) is used as a nonparametric prior in a hierarchical Bayesian setting. When applied to the clustering problem, the number of parameters in this model grows based on the complexity of the data. Let **GEM**(α) (**GEM** distribution is named after Griffiths-Engen-McCloskey) denote the stick-breaking distribution with parameter α. H is the global probability measure which is a discrete distribution. The definition of the model as shown in Figure 3.30 (a) is given below:

1. Draw a discrete distribution θ from **GEM**(α)
2. Draw a discrete distribution ϕ_k from $H(\beta)$
3. For each observed variable w_i^d in the document d
   (a) Draw the latent topic z_i^d from θ
   (b) Draw w_i^d from $F(\phi_{z_i^d})$

In the model shown in Figure 3.30, words are sampled from some parameterized family $F(\phi)$. Each observation w_i^d is based on an independently sampled parameter $\hat{\phi}_i^d$ (refer Figure 3.30 (b)). Therefore we note that:

$$\begin{aligned} \hat{\phi}_i^d &\sim G \\ w_i^d &\sim F(\hat{\phi}_i^d) \end{aligned} \tag{3.35}$$

In order to bring about higher flexibility and robustness in the Dirichlet process mixture model, a Dirichlet process prior, $G \sim \mathbf{DP}(\alpha, H)$, is placed on the latent parameter distribution. According to the stick-breaking construction:

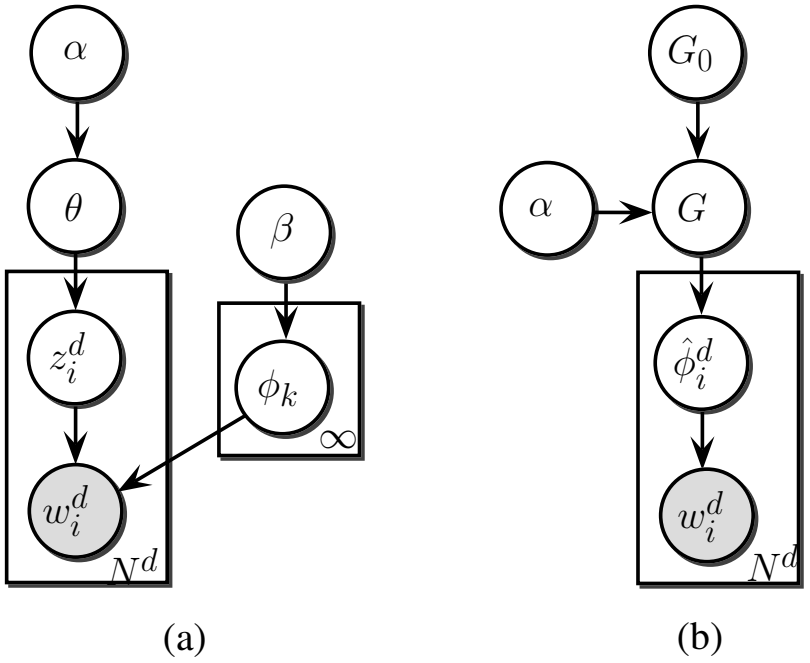


Figure 3.30: Directed graphical representation of an infinite Dirichlet process mixture model in a plate notation. The figure in (a) shows the stick breaking representation whereas in (b) shows an alternate distributional form.

$$G(\phi) = \sum_{k=1}^{\infty} \phi_k \delta(\phi, \phi_k)$$

$$\theta \sim \mathbf{GEM}(\alpha)$$

$$\phi_k \sim H(\beta), k = 1, 2, 3, \cdots$$

In this model, $F(\phi)$ is generally chosen as an exponential family of probability densities and $H(\beta)$ as its corresponding conjugate prior.

Let z_i^d denote the unique cluster index associated with w_i^d, the generative process of Equation 3.35 can be equivalently expressed (see Figure 3.30 (a)):

$z_i^d \sim \theta$

$w_i^d \sim F(\phi_{z_i^d})$

Marginalizing these indicator variables reveals an infinite mixture model with following form:

$$P(w_i^d|\theta, \phi_1, \phi_2, \cdots) = \sum_{k=1}^{\infty} \theta_k f(w|\phi_k) \tag{3.36}$$

The Pòlya-urn scheme is also known as CRP. In the CRP previously drawn values of ϕ have strictly positive probability of being re-drawn again. This makes the underlying probability measure G a discrete with probability one [68]. We obtain a Dirichlet process mixture model when we use a **DP** at the top of the hierarchical model.

3.10.4 Chinese Restaurant Process (CRP)

The CRP is a single parameter distribution over partitions of the integers CRP is a single parameter distribution over partitions of the integers [204]. The process is described by considering a Chinese restaurant with an infinite number of tables. N customers arrive in sequence and are labeled with integers $\{1, 2, \cdots, N\}$ and each customer sits down at a randomly chosen table. The first customer sits at the first table. After N customers have sat down, their configuration at the tables represent a random partition. The probability of a customer sitting at a table is computed from the number of other customers already sitting at that table. Let each z_i^d be distributed according to G. Let z_i^d denote the table assignment of the i^{th} customer and let us assume that the customers $z_{1:(i-1)}^d$ occupy K tables. Let n_k be the number of customers currently sitting at the table k. The n^{th} customer then sits at the table k with probability $\frac{n_k}{\alpha+n-1}$ and a new table is drawn with probability $\frac{\alpha}{\alpha+n-1}$. More formally, this is written as:

$$P(\text{occupied table } i|\text{previous customers}) = \frac{n_k}{\alpha + n - 1} \tag{3.37}$$

$$P(\text{next occupied table}|\text{previous customers}) = \frac{\alpha}{\alpha + n - 1} \tag{3.38}$$

The conditional distribution of the successive distribution of z_i^d given $z_1^d, z_2^d, \cdots, z_{i-1}^d$ where G has been integrated out is given by:

$$z_i^d | z_1^d, z_2^d, \cdots, z_{i-1}^d, \alpha, H \sim \sum_{l=1}^{i-1} \frac{1}{i-1+\alpha} \delta_{z_l^d} + \frac{\alpha}{i-1+\alpha} \tag{3.39}$$

Let $z_1^d, z_2^d, \cdots, z_{i-1}^d$ take on distinct values $\phi_1, \phi_2, \cdots, \phi_K$ and let m_k be the number of $z_{\hat{i}}^d$ that are equal to ϕ_k for $1 \leq \hat{i} < i$. Equation 3.39 can then be written as:

$$z_i^d | z_1^d, z_2^d, \cdots, z_{i-1}^d, \alpha, H \sim \sum_{k=1}^{K} \frac{m_k}{i-1+\alpha} \delta_{\phi_k} + \frac{\alpha}{i-1+\alpha} \tag{3.40}$$

3.10.5 Hierarchical Dirichlet Processes

In order to circumvent the limitation prevalent in parametric topic models such as the LDA model, Teh et al. [247] proposed the HDP model depicted in Figure 3.31. This model can be regarded as a nonparametric version of the LDA model [56]. HDP can be regarded as a nonparametric Bayesian model which is a Bayesian model on an ∞-dimensional parameter space. For nonparametric models, the number of parameters grow with the sample size. Incorporating the hierarchical nature in the DPMM introduces a mixed-membership property in which sharing among the clusters exist. This sharing among the clusters brings out a variety of relationships among the clusters in the topic space [215]. Inquisitive readers are requested to consult [247], [70], [239] for more details.

Given a collection of the text documents, HDP is characterized by a set of random probability measures G_d for each document d in the collection. In addition, a global random probability measure G_0 which itself is drawn from a Dirichlet Process (**DP**) with the base probability measure H. The global measure G_0 selects all the possible topics from the base measure H, and then each G_d draws the topics necessary for

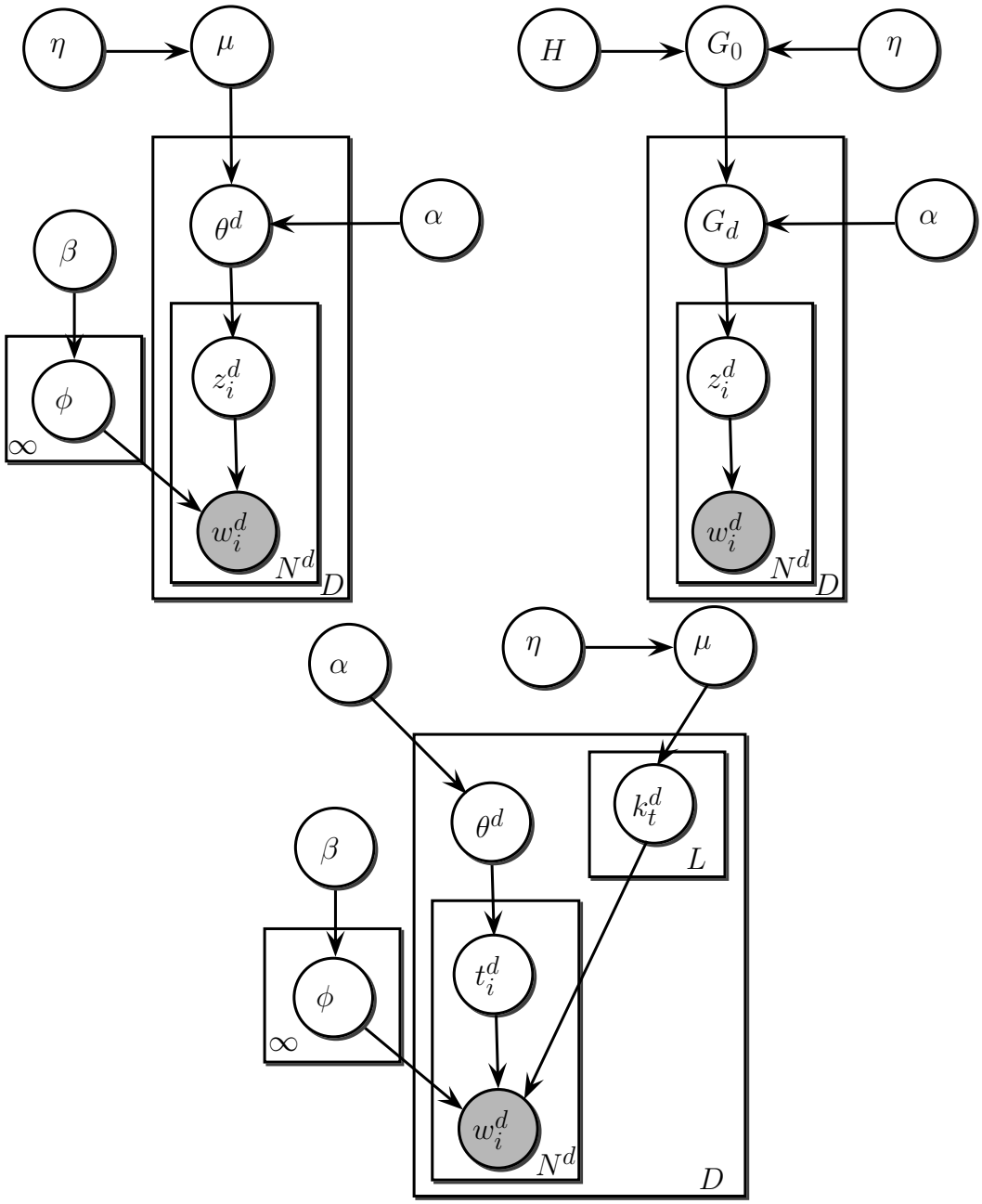


Figure 3.31: The depiction of the HDP model in three different schemes. The one on the top left is in the stick-breaking construction, and the top right figure depicts the HDP model in an alternative distributional form. The figure on on the bottom depicts in the Chinese Restaurant Franchise scheme.

the document d from G_0. The generative process of the model shown in Figure 3.31 (c) is described as:

1. Draw global probability measure G_0 from $\mathbf{DP}(\eta, H)$
2. For each document d in the collection
   (a) Draw G_d from $\mathbf{DP}(\alpha, G_0)$
   (b) For each word w_i^d
      i. Draw z_i^d from G_d
      ii. Draw w_i^d from $\mathbf{Multinomial}(z_i^d)$

where η and α are the concentration parameters that governs the variability around G_0 and G_d respectively. The base probability measure H provides the prior distribution for the factors z_i^d. Each z_i^d is a factor corresponding to a single observation w_i^d which is the word at the position i in the document d. The model in this distributional form is depicted in Figure 3.31 (c).

The HDP model is constructed by first sampling a global probability measure G_0 from Dirichlet process ($\mathbf{DP}(\eta, H)$) which defines a set of shared clusters. This is shown in Figure 3.31 (c). In the stick-breaking construction, which is another metaphor for the HDP process and shown in Figure 3.31 (a), is a constructive definition of a **DP**-distributed random distribution where we assume an infinite number of a unit-length sticks. They are then imagined to be broken off successively. In order to determine from where the stick needs to be broken, a random variable $G_0(\phi)$ is sampled from a Beta distribution, which is written as:

$$G_0(\phi) \sim \text{Beta}(1, \eta) \tag{3.41}$$

where η is a parameter that determines the peakiness of the distribution. Each ϕ_z is drawn from a base distribution $H(\beta)$. This base distribution could be either discrete or continuous. When $\mu \sim \text{GEM}(\eta)$, each ϕ_z is drawn from $H(\beta)$ which are independently and identically distributed draws from the base measure H. Note that H is depicted in Figure 3.31 (c). δ_{ϕ_z} is an atom which is a point distribution at ϕ_z and $z \in \{1, \cdots, \infty\}$. Then the following is a discrete random distribution:

$$G_0(\phi) = \sum_{z=1}^{\infty} \mu_z \delta(\phi, \phi_z) \tag{3.42}$$

In this way, each $G_0(\phi) \in (0, 1)$. A piece of the unit-length stick is then broken off at $G_0(\phi_z)$ and this process continues where the remainder of the stick is scaled to unit-length and another $G_0(\phi_z)$ is drawn from the distribution and thus the stick is broken again. It has been shown in [229] that Equation 3.42 is distributed according to $\mathbf{DP}(\alpha, H)$ and thus μ_z and ϕ_z can be marginalized out, G is called the Dirichlet Process (**DP**) [247].

The stick breaking representation is given as:

1. Draw μ from $\mathbf{GEM}(\eta)$
2. Draw ϕ_z from $H(\beta)$
3. For each document d in the collection
   (a) Draw θ^d from $\mathbf{DP}(\alpha, \mu)$
   (b) For each word w_i^d in the document d
      i. Draw z_i^d from θ^d
      ii. Draw w_i^d from $\mathbf{Multinomial}(\phi_{z_i^d})$

One perspective associated with the HDP mechanism can be expressed by the CRF [247] which is an extension of the CRP [3]. We depict this model in a standard plate notation in Figure 3.31 (b). In order to describe the sharing among the groups, the notion of "franchise" has been introduced that serves the same set of dishes globally. When applied to text data, each restaurant corresponds to a document. Each customer corresponds to a word. Each dish corresponds to a topic. A customer sits at a table, one dish is ordered for that table and all subsequent customers who sit at that table share that dish. The dishes are sampled from the base distribution H which corresponds to discrete topic distributions. Multiple tables in multiple restaurants can serve the same dish. The factor values are shared both between and amongst documents. For a complete mathematical derivation of the CRF metaphor, we direct the reader to review [247]. As mentioned before, one major limitation of the HDP model is that it loses the document's structural information related to word ordering, and as a result it generates only unigram words in topics which may not convey much insight for user interpretation of a topic.

The generative process in the Chinese Restaurant Franchise scheme is given as:

1. Draw μ from $\mathbf{GEM}(\eta)$
2. Draw ϕ_z from $H(\beta)$
3. For each document d
   (a) Draw θ^d from $\mathbf{DP}(\alpha, \mu)$
   (b) Draw k_t^d from μ
   (c) For each word w_i^d at position i in the document d
      i. Draw t_i^d from θ^d
      ii. Draw w_i^d from $\phi_{k_{t_i^d}^d}$

In [247], apart from the incremental Gibbs sampling scheme, the author also proposed two more sampling schemes to reduce significant book-keeping effort. One inference scheme is based on Augmented Representation and the other describes the Direct Assignment scheme. The Direct Sampling scheme significantly eases the implementation of the inference algorithm, whereas this scheme changes the component membership in each iteration one at a time. But the other two schemes change the memberships of multiple data items because changing the component membership of one table changes the memberships of all the data items associated with that table. Therefore we expect the Direct Assignment scheme to be much slower in convergence than the other two schemes. In [247], it provides a complete derivation of posterior sampling using Augmented Representation and Direct Assignment.

3.11 Supervised Topic Models

3.12 MedLDA Model

The prime motivation for designing the MedLDA [297] model is that the model is able to apply more discriminative maximum-margin learning technique within a probabilistic framework than the existing methods. The model makes use of extra information (also called as the side information) in order to improve the prediction task, and to generate more interpretable topics. MedLDA integrates the power of a discriminative model along with a generative model under a unified constrained optimization framework. The resulting model apart from finding a low-dimensional representation of the original vector space, is also able to separate the latent topical clusters that are more discriminative.

The MedLDA model can be readily applied to the other problem tasks in addition to text data. The graphical model of the MedLDA model is shown in Figure 3.32. The

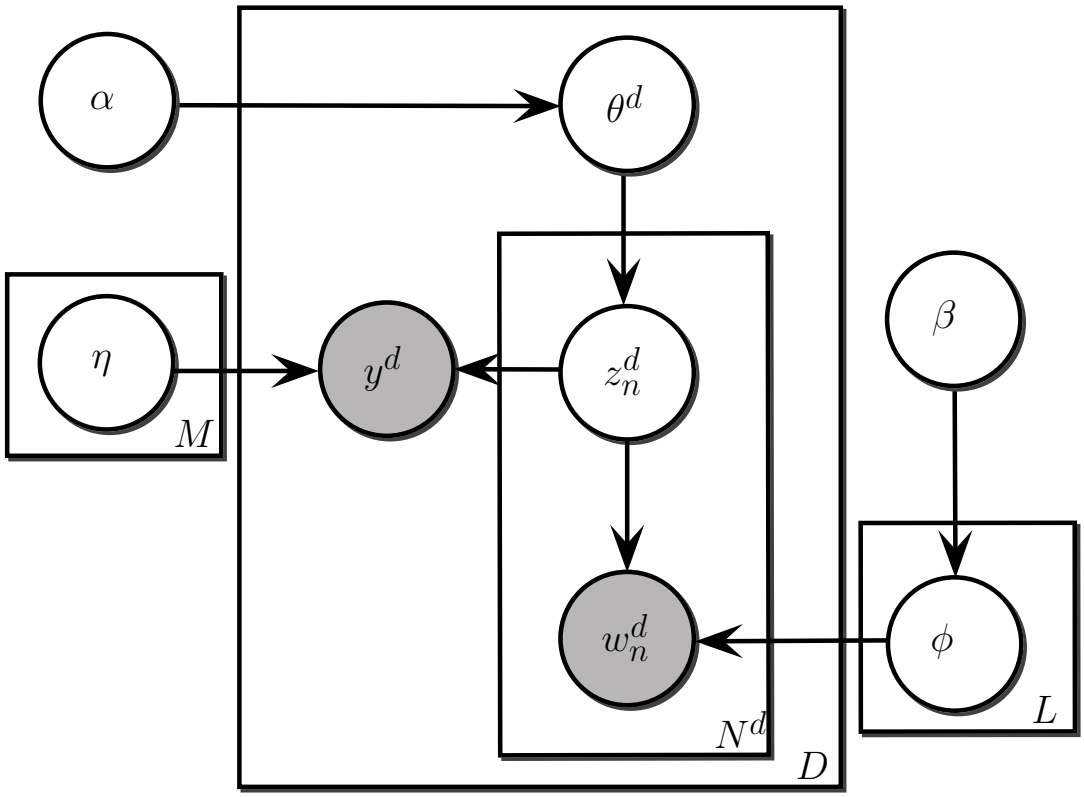


Figure 3.32: The graphical model of the `MedLDA` model in plate notation. The models depicts that order of words in the document is not important. Different from the `LDA` model, we see that this model incorporates an observed variable y^d for each document. This observed variable a side information or response variable which is used during the inference step.

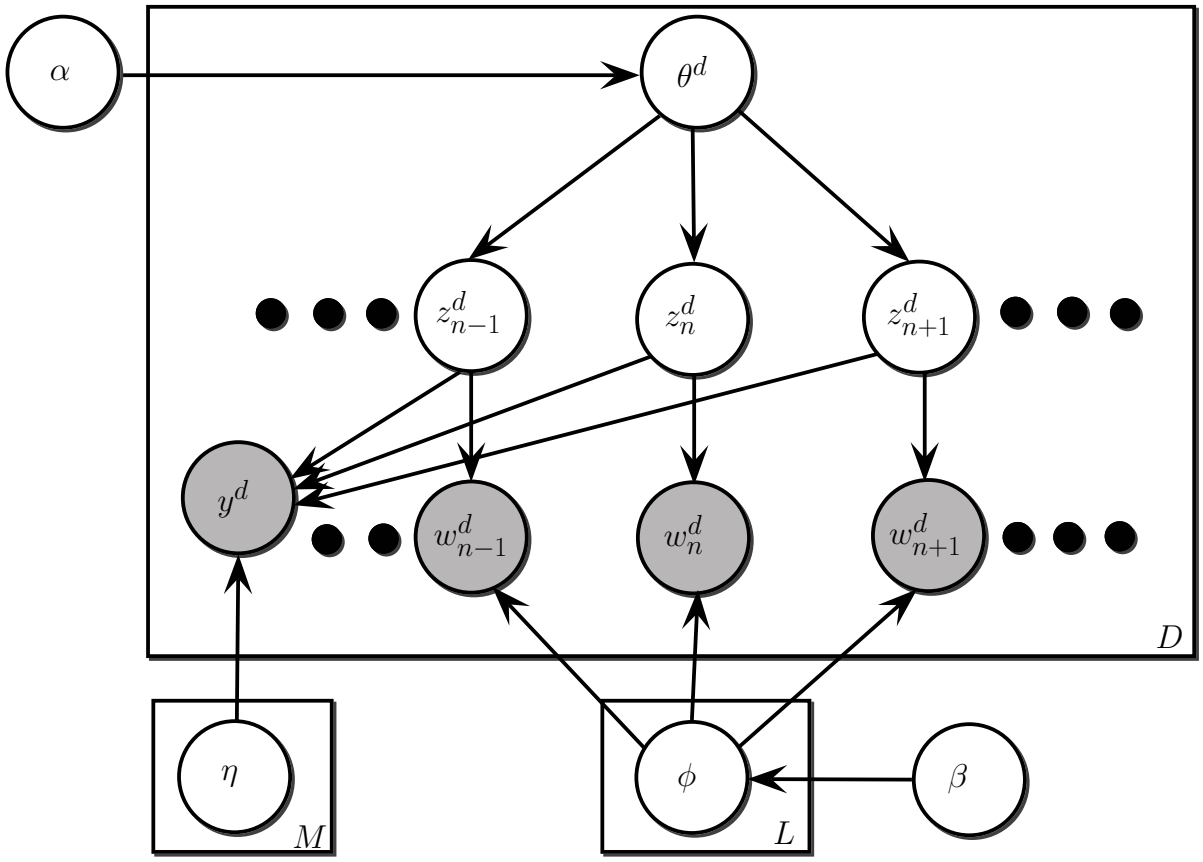


Figure 3.33: The graphical model of the `MedLDA` model in expanded plate notation. This helps us show the words in the document, but the order of the words in the document is not maintained. Different from the `LDA` model, we see that this model incorporates an observed variable y^d for each document. This observed variable a side information or response variable which is used during the inference step.

graphical model in Figure 3.32 shows the compact structure of the model, whereas in Figure 3.33 (b), the model is expanded, and the words and their topical assignments are shown. It is noticeable that the order of the words is not considered in the MedLDA model. This in turn results in losing an important document structure, which may help the model improve in many text mining applications [117].

The MedLDA model as depicted in Figure 3.32 comprises of two models to form a hybrid model. One model is the LDA model shown in Figure 3.10 (which considers the document contents) and the other is a maximum-margin prediction classifier which considers the supervision labels. LDA is an unsupervised method to find a low-dimensional latent structure inherent in the data, whereas the maximum-margin approach such as the SVM [53] is a supervised learning approach which requires an initial set of training data with tagged information given by some oracle. Although the graphical model of MedLDA matches with that of the sLDA [21], there are many key differences in the way two of them model the data [156], [157]. One of the key differences is that the MedLDA model imposes a discriminative constraint directly on the posterior distributions [123].

As depicted in Figure 3.32, the model assumes independence among the latent variables $\boldsymbol{z}$. The response variable Y attached with each document incorporates the supervised side information consisting of certain class labels. The maximum margin principle is directly used to generate Y. Maximum margin learning and Maximum likelihood estimation is both used while learning the parameters of the model. Thus two stages are involved. In the first stage unsupervised topic discovery is made, and then maximum margin multi-class classification is performed.

The generative process of the MedLDA model can be written as follows:

1. Draw ϕ_z from **Dirichlet** ($\boldsymbol{\beta}$) for each topic z, where ϕ_z is the word distribution for topic z and $z \in \{1, \ldots, L\}$. β is the parameter of the Dirichlet prior on the

per-topic word distribution.

2. For each document d

   (a) Draw a topic proportion θ^d for a document d where $d \in [1, \ldots, D]$ from **Dirichlet** (α), where **Dirichlet** (α) is the Dirichlet distribution with parameter α on the per-document topic distributions,

   (b) For each word w_i^d at position i in the document d

      i. Draw a topic z_i^d for each word w_i^d at position i in the document d from **Multinomial** (θ^d)

      ii. Draw a word w_i^d from **Multinomial** $(\phi_{z_i^d})$

3. Draw the response parameter $\boldsymbol{\eta}$ from **Normal** $(0, \boldsymbol{\eta}_0)$, where $\boldsymbol{\eta}_0$ is the hyper-parameter for $\boldsymbol{\eta}$ and is sampled M times

4. Draw a response variable $y^d | \boldsymbol{z}^d, \boldsymbol{\eta}$ from $F(y^d, \boldsymbol{z}^d, \boldsymbol{\eta}) = \boldsymbol{\eta}_y^\intercal \overline{\boldsymbol{z}}$, where y^d is a class-specific vector associated with class y. $\boldsymbol{\eta}_y$ is a class specific K dimensional vector associated with class y. $\overline{\boldsymbol{z}} = \frac{1}{N^d} \sum_{n=1}^{N^d} z_n^d$.

The MedLDA model can be used for both regression and classification. However, we will limit our discussion to the classification task in this paper. Subsequently, the MedLDA model will be extended to incorporate word-order. We will then show how the model can be used in ranking of text documents. The original paper on MedLDA [298], [297] contains the posterior inference description using the variational methods, but our focus will be mainly on using Gibbs sampling for our model and also we will focus on using Gibbs sampling procedure for the MedLDA model which has been proposed recently in [299], [123].

As shown in Figures 3.32 and 3.33, the MedLDA model comprises of two parts. One of the parts is the LDA model and the other part is a classifier built on taking the

expectation of all the models and is mainly an averaging model under the Bayesian paradigm. We will present a short description of both the models separately and then describe a hybrid regularized Bayesian model which is a combination of a maximum margin approach and the LDA model.

Suppose $T = \{(\boldsymbol{w}_d, y_d)\}_{d=1}^{D}$ be a given fully-labeled training set. Since we consider a multi-class classification problem, the values of the response variable can come from a finite set $Y = \{(1, \ldots, M)\}$. We will present brief description of the each of the components in the MedLDA model.

One of the components is the LDA model, which we have already described in Section 3.7. We present the description of few content which will be used in the MedLDA model for further derivation.

When the parameters of the model are given, the joint distribution of a topic mixture θ^d, a set of topics $\boldsymbol{z}^d$ for that document d, and the words in that document $\boldsymbol{w}^d$ is given by:

$$P(\theta^d, \boldsymbol{z}^d, \boldsymbol{w}^d|\alpha, \beta) = P(\theta^d|\alpha) \prod_{i=1}^{N^d} P(z_i^d|\theta^d) P(w_i^d|z_i^d, \beta) \tag{3.43}$$

By taking the product of the marginal probabilities of all the documents in the collection T, the probability for the entire collection can be written as:

$$P(T|\alpha, \beta) = \prod_{d=1}^{D} \int P(\theta^d|\alpha) \left(\prod_{i=1}^{N^d} \sum_{z_i^d} P(z_i^d|\theta^d) P(w_i^d|z_i^d, \beta) \right) \mathrm{d}\theta^d \tag{3.44}$$

Computing the exact posterior distributions in the LDA model is intractable. Hence one has to resort to approximation techniques. Methods such as variational inference [23], Gibbs sampling [86], collapsed Gibbs sampling [208] and many other techniques have been used in order to compute an approximation. The posterior

distribution inferred by the LDA model is written as:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \alpha, \beta) = \frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\alpha, \beta)} \tag{3.45}$$

where $P_0(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}) = (\prod_{d=1}^{D} P(\theta^d|\alpha) \prod_{n}^{V} P(z_n^d|\theta^d)) \prod_{k=1}^{K} P(\phi_k|\beta)$ is the joint distribution defined in the model. In [287], the author has studied that the posterior distribution by Bayes' rule is the solution to an optimization problem. Therefore, the posterior distribution shown in Equation 3.45 can be transformed to an optimization problem which can be written as:

$$\begin{aligned} \underset{P(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}} \quad & \mathrm{KL}[P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)] - \mathbb{E}_P[\log \mathrm{P}(\boldsymbol{W}|\boldsymbol{Z}, \boldsymbol{\Phi})] \\ \text{subject to} \quad & P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}, \end{aligned} \tag{3.46}$$

where $\mathbb{P}$ is the probability distribution space, and $\mathrm{KL}(P||P_0)$ is the Kullback-Leibler divergence from P to P_0. We present the full proof of the above equation in Appendix E[6]

The MedLDA model also comprises of an expected classifier. The description of the classification component is as follows:

Consider a set of training documents T. The task of the classifier is to incorporate the lowest possible risk which is approximated by the training error. The posterior distribution $P(h|T)$ is selected by the classifier over a hypothesis space H comprising of classifiers in such a way that the P-weighted classifier $h_p(\boldsymbol{w}) = \mathrm{sign}\mathbb{E}_P[h(\boldsymbol{w})]$ has the lowest possible risk. The posterior distribution $P(\eta, \boldsymbol{\Theta}, Z, \boldsymbol{\Phi}|T)$ that needs to be computed by the classifier of the MedLDA model, should have the lowest possible risk. This classifier for the MedLDA model can be written as:

$$\hat{y} = \mathrm{sign}F(\boldsymbol{w}) \tag{3.47}$$

[6]References were made to ascertain the correctness of the proof from here http://mark.reid.name/blog/bayesian-updating-as-optimisation.html

The risk is approximated by the training error:

$$T_T(P) = \sum_d \mathbb{I}(\hat{y}^d \neq y^d) \tag{3.48}$$

The discriminant function can be written as:

$$F(\boldsymbol{w}) = \mathbb{E}_{P(\boldsymbol{\eta},\boldsymbol{z}|T)}[F(\boldsymbol{\eta},\boldsymbol{z};\boldsymbol{w})], F(\boldsymbol{\eta},\boldsymbol{z};\boldsymbol{w}) = \boldsymbol{\eta}^\intercal \overline{\boldsymbol{z}} \tag{3.49}$$

where $\overline{\boldsymbol{z}}$ is a vector with element $\overline{z}_k = \frac{1}{N^d}\sum_{n=1}^{N^d} \mathbb{I}(z_{nk} = 1)$. $\mathbb{I}(.)$ is an indicator function which equals to 1 if predicate holds else it is 0.

$$\begin{aligned} \underset{p(\boldsymbol{\eta},\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P},\psi}{\text{minimize}} \quad & \mathrm{KLD}[q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})] - \mathbb{E}_q[\log \mathrm{P}(\boldsymbol{W}|\boldsymbol{Z},\boldsymbol{\Phi})] + \frac{C}{D}\sum_{d=1}^{D}\psi^d \\ \text{subject to} \quad & \mathbb{E}_p[\boldsymbol{\eta}^\intercal \boldsymbol{f}(y^d,\overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y,\overline{\boldsymbol{z}}^d)] \geq l^d(y), \psi^d \geq 0, \forall d, \forall y, \end{aligned} \tag{3.50}$$

Equation 3.50 can also be written as:

$$\begin{aligned} \underset{P(\boldsymbol{\eta},\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}} \; & \mathrm{KL}[q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})] - \mathbb{E}_q[\log \mathrm{P}(\boldsymbol{W}|\boldsymbol{Z},\boldsymbol{\Phi})] + \\ & \frac{C}{D}\sum_d \mathrm{argmax}_x(l^d(y)) - \mathbb{E}_p[\boldsymbol{\eta}^\intercal]\boldsymbol{f}(y^d,\overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y,\overline{\boldsymbol{z}}^d) \end{aligned} \tag{3.51}$$

The component $\frac{1}{D}\sum_d \mathrm{argmax}_x(l^d(y)) - \mathbb{E}_p[\boldsymbol{\eta}^\intercal]\boldsymbol{f}(y^d,\overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y,\overline{\boldsymbol{z}}^d)$ is the hinge loss which is defined as an upper bound of the prediction error on the training data.

3.12.1 Posterior Inference using Gibbs Sampling

Computing the exact posterior distribution in topics models is an intractable problem. Hence approximation methods have been commonly used such as variational methods and Markov Chain Monte Carlo (MCMC). We will describe a collapsed Gibbs sampling method in this paper. In order to begin with the MCMC method, we first need to assume that:

$$P(\boldsymbol{\eta}, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) = P(\boldsymbol{\eta}) \times P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \tag{3.52}$$

Thus what is required now is to alternately solve the Equation 3.51 in the following two steps:

Step 1: Estimation of $P(\boldsymbol{\eta})$, this problem can be written as a constrained form as:

$$\begin{aligned} \underset{P(\boldsymbol{\eta},\boldsymbol{\psi})}{\text{minimize}} \quad & \text{KLD}(P(\boldsymbol{\eta})||P_0(\boldsymbol{\eta})) + \frac{C}{D}\sum_{d=1}^{D}\psi^d \\ \text{subject to} \quad & \mathbb{E}_p[\boldsymbol{\eta}^\intercal \boldsymbol{f}(y^d, \overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y, \overline{\boldsymbol{z}}^d)] \geq l^d(y), \psi^d \geq 0, \forall d, \forall y, \end{aligned} \tag{3.53}$$

The optimum posterior distribution can be computed by using the Langrangian methods with multipliers $\boldsymbol{\lambda}$, as follows:

$$P(\boldsymbol{\eta}) \propto P_0(\boldsymbol{\eta}) e^{\boldsymbol{\eta}^\intercal \cdot \sum_D^{d=1} \sum_y \lambda_y^d \boldsymbol{\Delta f}(y, \mathbf{E}[\overline{\boldsymbol{z}}^d])} \tag{3.54}$$

As described in the generative process for the MedLDA model, we chose the standard normal prior $P_0(\boldsymbol{\eta} = N(0, I))$. In the case of the MedLDA model, the prior can

be written as $P(\boldsymbol{\eta}) = N(\boldsymbol{\kappa}, I)$, the dual of the problem then becomes:

$$\begin{aligned} \underset{\boldsymbol{\lambda}}{\text{maximize}} \quad & \frac{-1}{2}\boldsymbol{\kappa}^\intercal\boldsymbol{\kappa} + \sum_{d=1}^{D}\sum_{y}\lambda_y^d l^d(y) \\ \text{subject to} \quad & \sum_{y}\lambda_y^d \in [0, \frac{C}{D}], \forall d \end{aligned} \tag{3.55}$$

where $\boldsymbol{\kappa} = \sum_{d=1}^{D}\sum_{y}\lambda_y^d \boldsymbol{\Delta f}(y, \mathbb{E}[\overline{\boldsymbol{z}}^d])$, where $\boldsymbol{\kappa}$ is the mean of classifier parameters $\boldsymbol{\eta}$. The element κ_{yk} represents the contribution of topic k in classifying a data point to category y.

Step 2: Estimation $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$: Given $P(\boldsymbol{\eta})$, this subproblem can be resolved to solve the following:

$$\begin{aligned} \underset{p(\boldsymbol{\eta},\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P},\psi}{\text{minimize}} \quad & \text{KLD}[q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] - \mathbb{E}_q[\log \text{P}(\boldsymbol{W}|\boldsymbol{Z}, \boldsymbol{\Phi})] + \frac{C}{D}\sum_{d=1}^{D}\psi^d \\ \text{subject to} \quad & (\boldsymbol{\kappa}^*)^\intercal\boldsymbol{\Delta f}(y, \mathbb{E}_P[\overline{\boldsymbol{z}}^d]) \geq l^d(y) - \psi^d, \psi^d \geq 0, \forall d, \forall y, \end{aligned} \tag{3.56}$$

Equation 3.56 can be further written as which will be used in for updating the posterior estimates:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \propto P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}, \boldsymbol{W}) e^{\boldsymbol{\kappa}^{(*)\intercal}\sum_{d=1}^{D}\sum_{y}(\lambda_y^d)^*\boldsymbol{\Delta f}(y,\overline{\boldsymbol{z}}^d)} \tag{3.57}$$

In Appendix F, we present the full derivation of the collapsed Gibbs sampling for the MedLDA model.

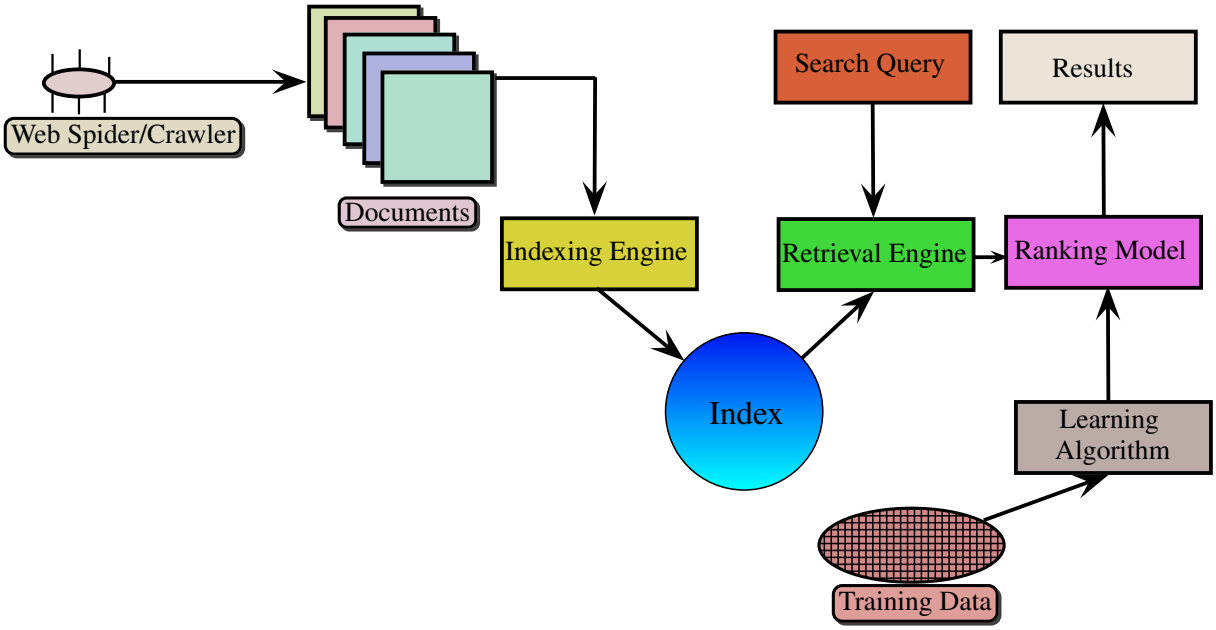


Figure 3.34: A figure depicting the high-level architecture of the learning-to-rank framework. The web crawler crawls the data, which is then indexed by the indexer. Then some subset of the data is used to learn a ranking model which is then used to rank unseen document-query pairs.

3.13 Learning-to-Rank

Ranking is an important problem in IR. Using machine learning techniques to learn a ranking function has shown some promise in the recent past. Learning-to-rank has been applied to a vast range of problem domains such as IR [167], data mining [268], Natural Language Processing (NLP) [158], etc. Learning-to-rank sometimes also called as machine learned ranking can be characterized as a supervised, unsupervised, and semi-supervised based learning, where the objective is to construct a ranking function for IR. In this process, the training data consists of a list of items, in particular, query-document pairs along with the corresponding relevance information/judgment, which is obtained using some external process such as human generated annotation process, with some partial order specified between items in each list. The objective is then to rank a list of unseen data which is Independent and identically distributed (IID) as the training data.

In Figure 3.34, we depict a high-level architecture diagram of the learning-to-rank framework. The web crawler crawls web pages from the web space, which is then passed onto an indexer which does preprocessing and indexing of the web pages. Then the index is stored in the local storage for conducting retrieval. Some subset of the documents in the index can be manually annotated to get relevance assessments from humans which can be used to learn a ranking function. This learned ranking model can then be used to re-rank the results of the top-k results based on the query entered by the user.

3.13.1 Features

The features used to train the learning-to-rank models consist of both high-level and low-level features. In fact, there is an extensive list of features that is currently in use in learning-to-rank algorithms today. In order to study more about those extensive set of features, one is requested to consult [210] for more details.

Chapter Four

Topic Segmentation Model for Text Documents

Chapter Summary

In this chapter, we present a new unsupervised topic discovery model, called N-gram Topic Segmentation model (`NTSeg`), for a collection of text documents. `NTSeg` maintains the structure of the document such as paragraphs and sentences. In addition, it preserves the word order in the document. `NTSeg` can help capture major topical changes in the document. As a result, it can generate two levels of topics of different granularity, namely, segment-topics and word-topics. In addition, it can generate n-gram words in each topic.

4.1 The Case for Topic Segmentation with Word Order

As we have studied in Chapters 1, 2 and 3 that topic models such as the LDA [23] have been widely used to find topics in a document collection. But the LDA model has been criticized for its bag-of-words assumption [252] as the model does not consider the structural information inherent in the text which could help tap extra knowledge from the text. It is well known that the bag-of-words assumption is mainly a simplifying assumption to reduce the complexity of the model [179], [115].

Some previous works demonstrate that considering the ordering of words is desirable [261], [252]. Maintaining the word order during the processing of documents introduces some computational overhead, but it allows us to achieve what the bag-of-words models cannot do in general [88], [117]. In order to address the shortcoming inherent in the LDA model, the authors in [261] introduced the topical n-gram model (TNG) to find n-gram words in topics. This model has the ability to decide whether to form a unigram or a bigram during topic discovery by extending the LDACOL model

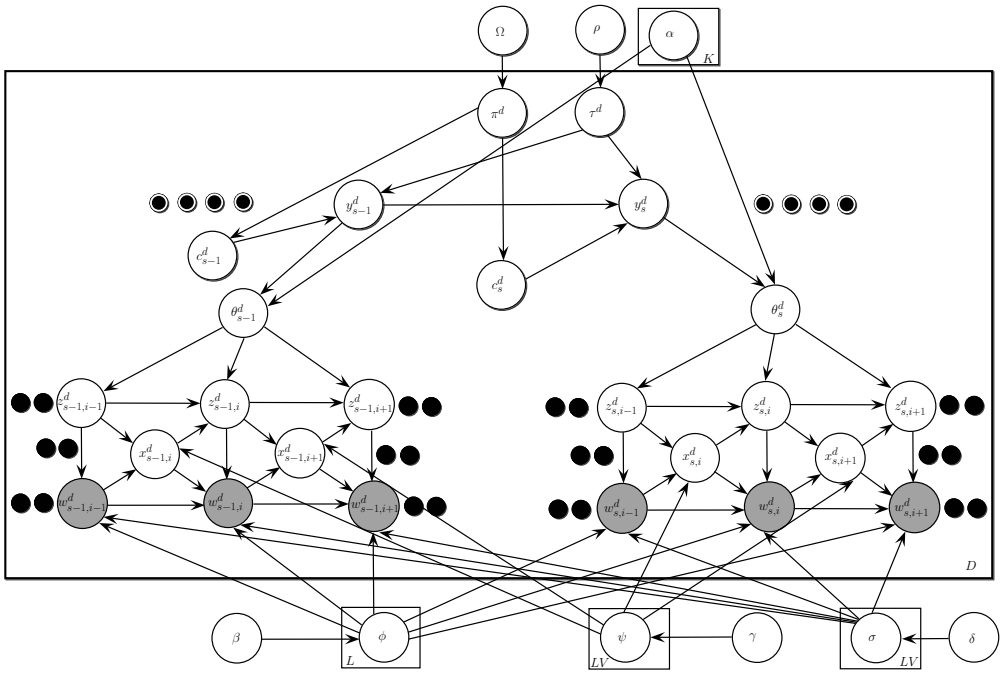


Figure 4.1: Our proposed `NTSeg` model for topic segmentation and topical n-gram word generation. The model is represented as a graphical model in standard plate notation where plates signify repetition of variables.

[87] and the `BTM` [252]. All these models advocate that the word order in a document is essential. But one shortcoming of these models is that they lack the ability to consider the document's structure such as paragraphs and sentences. Thus they cannot segment a document into coherent topics. This sometimes becomes essential in tasks such as tackling the word sense disambiguation problem as shown in [88], segmenting news articles and finding topics in each segment [220], topic detection and tracking [265], and a plethora of other tasks which motivate us to explore deeper into the topic segmentation model.

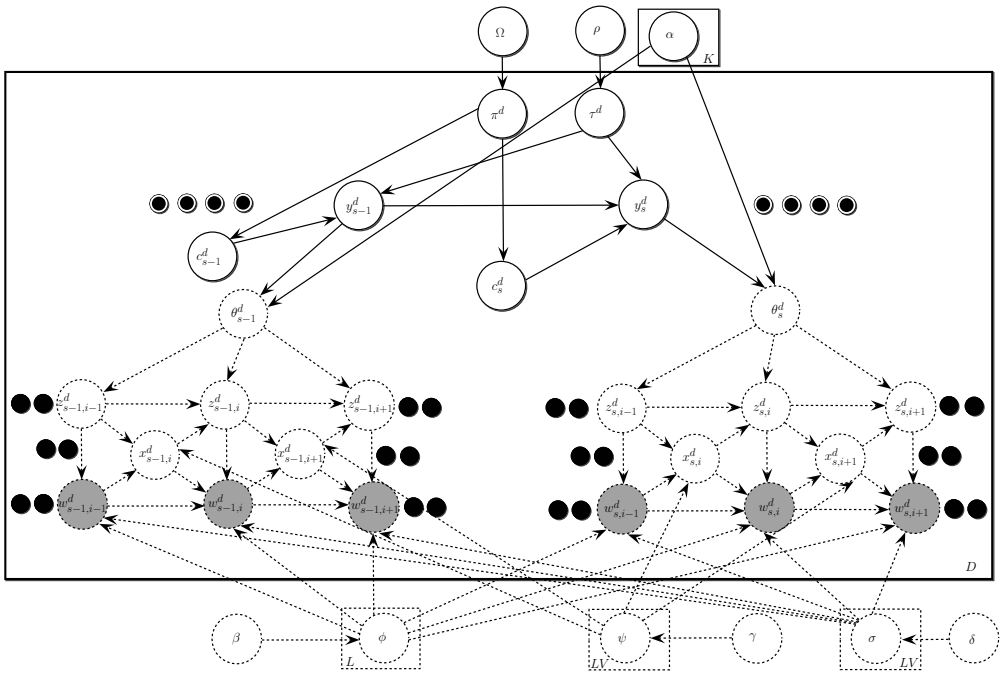


Figure 4.2: Our proposed `NTSeg` model for topic segmentation. The un-dotted section of the graphical model performs topic segmentation of text by generating segment-topics. The ordering of $\mathbf{c}$ in the document according to the document structure, portrays the segmentation of document.

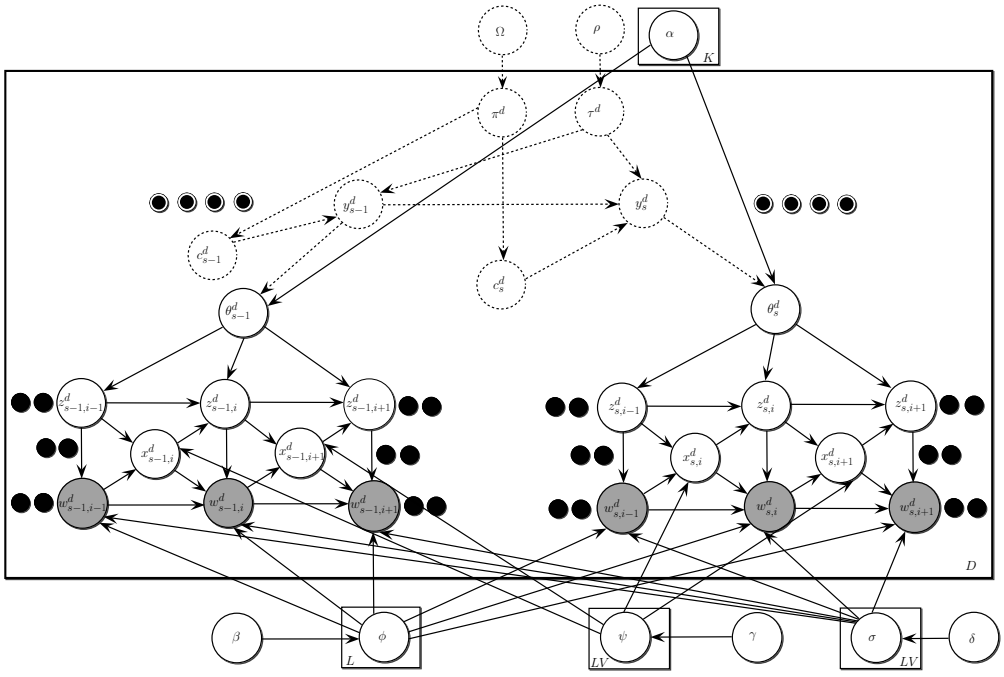


Figure 4.3: Our proposed `NTSeg` model for topic segmentation. The un-dotted section of the graphical model performs n-gram word generation where words share the same topic.

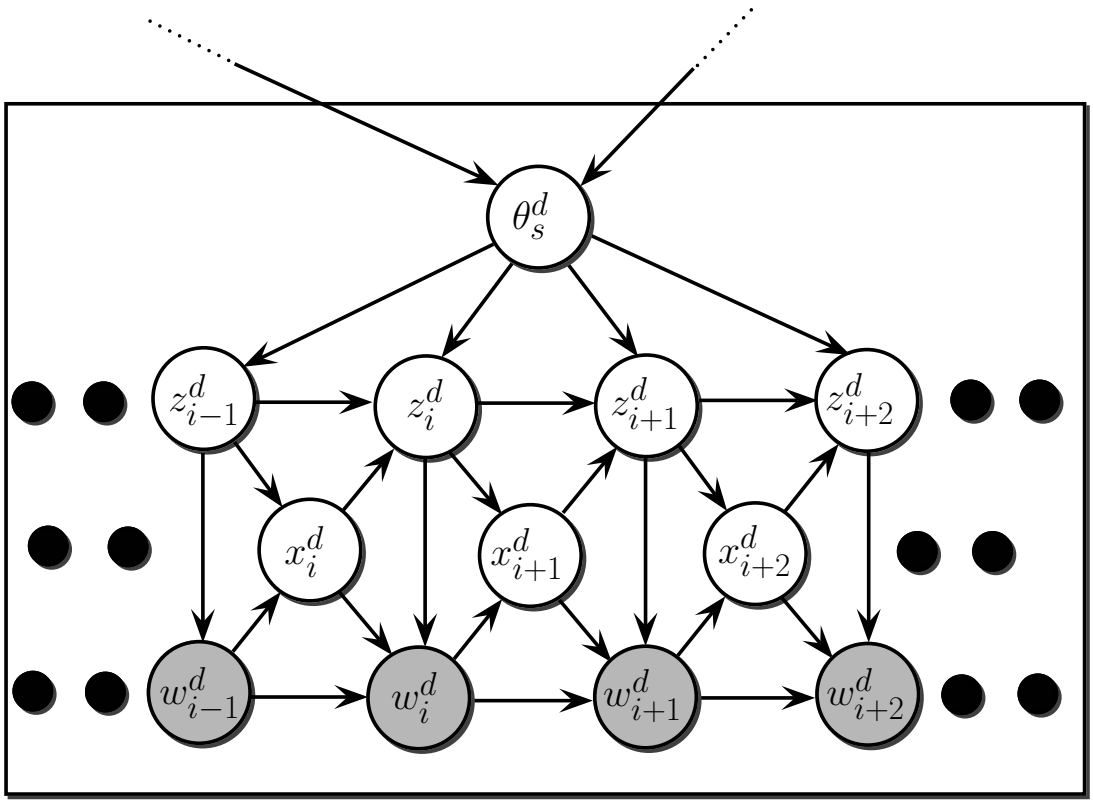


Figure 4.4: A component from the main graphical model shown in Figure 4.1, which depicts the n-gram word generation. We can see the the order of words is maintained, and words share the same topic.

4.2 N-gram Topic Segmentation Model Description

We depict our proposed NTSeg model in Figure 4.1 using a graphical model in a plate notation where shaded circles represent observed variables and unshaded ones are the latent variables. Plates signify repetition of the variables. The basic idea of our model is that a document comprises of several topically coherent segments such as paragraphs and sentences, and in each segment words occur in an order. Our model preserves this structure. In addition, it preserves the ordering of the words in order to capture the collocation and its semantic information. Thus NTSeg is no longer invariant to the reshuffling of the words in each segment.

For a properly written discourse comprehension, documents are generally composed of coherent segments which are semantically linked to one another so that a reader could relate the storyline as one moves forward in the discourse [140]. Our model comprises of two levels of topic of different granularity. One is the segment-topic to which segments in the documents are assigned, so that their ordering defines the major topical changes in the document. In Figure 4.2, we depict the portion of our model that performs topic segmentation i.e. the un-dotted lines. The other is the word-topic to which n-gram words in the segment are assigned. In Figure 4.3, we present the portion of the model which generates word-topics. In Figure 4.4, we further narrow down to the portion of the graphical model that generates word-topics, and where the order of words is maintained. The segment-topics come from a predefined number of segment-topics K . Each segment-topic comprises of a mixture of several word-topics where the mixture coefficients uniquely specify the segment-topic. Word-topics come from a predefined number of word-topics L . In general, the number of segment-topics will be less than the number of word-topics. The reason is that the number of segments (paragraphs or sentences) in a document is less than compared to the number of words [231].

From the graphical model in Figure 4.1, one can note that our model, NTSeg, has the capability of deciding whether to generate a unigram or a bigram in a topic, and the topic assignment for the words in a bigram are the same. This aspect differentiates NTSeg from TNG. NTSeg assumes a first order Markov assumption i.e. it is mainly a bigram model but the basic generation process produces unigram or bigrams. However, NTSeg has the ability to produce higher order n-grams (i.e. $n > 2$) by concatenating consecutive n-grams (unigrams or bigrams) having the same topic and the bigram status variable between them is 1. In this way, the words in the n-gram share the same topic. In Figure 4.5, we illustrate the idea using a diagrammatic representation, where we generate a tri-gram "Irish cricket team". We see in the figure that $x = 1$ between the words in sequence, and these words can then be concatenated together to form a tri-gram. In Figure 4.6, we exemplify

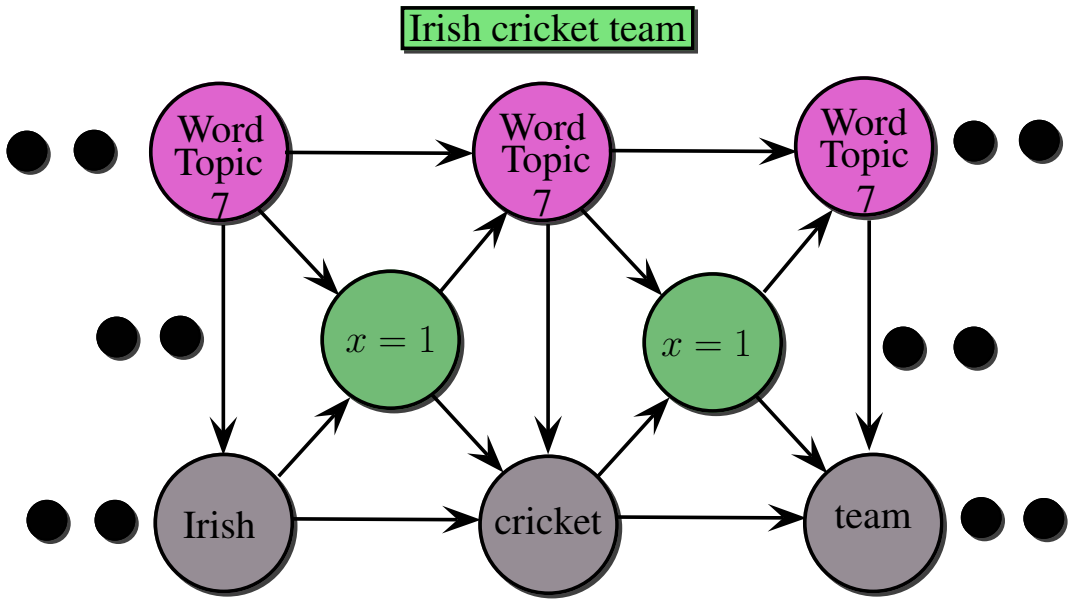


Figure 4.5: An example showing how a tri-gram "Irish cricket team" is generated by our model, where words in the tri-gram share the same word-topic which in this case is word-topic number 7. We can see that words in sequence are only concatenated in sequence when they share the same word-topic and when the bigram binary status random variable x between them is set to 1.

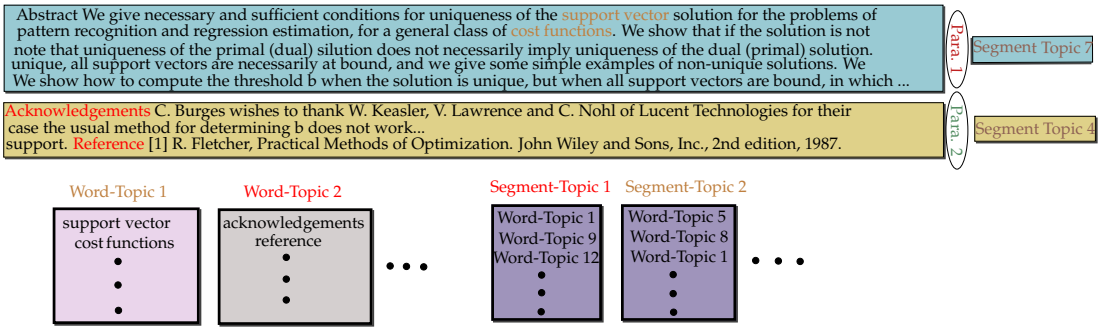


Figure 4.6: An example illustrating the word-topic and segment-topic allocation paradigm. Paragraphs are generated and assigned to the segment-topics, whereas n-gram words are generated by the word-topics. The figure shows that our model generates topics in a hierarchy, that is segment-topics consist of a mixture of word-topics. Word-topics generate n-grams.

the idea of topic hierarchy generated by our model. Our proposed model NTSeg generates segment-topics, and each segment-topic comprises of a mixture of word-topics. Word-topics consist of n-gram words instead of just unigram. We see that paragraphs are assigned to the segment-topics in the figure. This again contrasts NTSeg from TNG where TNG analyzes each topical n-gram post hoc as if the topic of the final word in the n-gram was the topic assignment of the entire n-gram. But it violates the principle of non-compositionality [166]. In each segment $s \in S$, NTSeg finds n-gram words in a word-topic $\mathbf{z}$. It can also find correlations between both kinds of topics i.e. word-topics $\mathbf{z}$ and segment-topics $\mathbf{y}$. The segments of each document are assumed to follow a Markov structure on the topic distributions of each segment. We assume that there will be a high probability that the topic for the segment s in the document will be the same as that of the segment $s-1$. A segment binary switching variable c_s^d for the segment-topic indicates whether there is a change of topic between the segments. The states of the switching variable correspond to the segmentation of the topic into coherent topical units. Apart from the segment switching binary variable, NTSeg also incorporates another random variable known as the bigram switch variable $\mathbf{x}$. The mechanism is that if $x_{si}^d = 1$, then w_{si-1}^d and w_{si}^d forms a bigram else it does not. Note that the input to NTSeg is the entire document.

It can be observed that the existing TNG model [261], LDSEG [230] and LDCC [231] are special cases derived from our model. For example removing the segmentation scheme in Figure 4.1 along with a set of arrows pointing from $z_{si-1}^d \rightarrow z_{si}^d$ and $x_{si}^d \rightarrow z_{si}^d$ reduces to the topical n-gram model. Removing the bag-of-words assumption in each segment reduces to the LDSEG model and relaxing both bag-of-words and removing the segmentation switch variable reduces to the LDCC model. NTSeg has the ability to decide whether to form a unigram or bigram based on context which LDSEG model cannot achieve.

The generative process of our model NTSeg for each document d is as follows:

1. Draw τ from **Dirichlet**(ρ) and ρ is the parameter of the Dirichlet prior on the segment-topics
2. Draw **Discrete**(ϕ_z) from **Dirichlet**(β) for each word-topic z
3. Draw **Bernoulli**(ψ_{zw}) from **Beta**(γ) for each word-topic z and each word w
4. Draw **Discrete**(σ_{zw}) from **Dirichlet**(δ) for each topic z and each word w
5. For each segment s from S
   (a) Draw y_s (i.e. the same segment-topic) for s as its previous segment topic y^d_{s-1} with probability $P(c_s = 1) = \pi$
   (b) Otherwise, draw a segment-topic for the segment y_s from **Multinomial**(τ)
   (c) Draw $\theta^{(s)}$ from **Dirichlet**(α, y_s)
   (d) For each N^d_s words in the segment w^d_{si}
      i. Draw x^d_{si} from **Bernoulli**($\psi_{z^d_{s(i-1)} w^d_{s(i-1)}}$)
      ii. Draw z^d_{si} from **Discrete**($\theta^{(s)}$) if $x^d_{si} = 0$ else draw w^d_{si} from the same topic as $w^d_{s(i-1)}$
      iii. Draw w^d_{si} from **Discrete**($\sigma_{z^d_{si} w^d_{s(i-1)}}$) if $x^d_{si} = 1$
      iv. Otherwise, Draw w^d_{si} from **Discrete**($\phi_{z^d_{si}}$)

c^d_s indicates whether there is a change in topic between the segments s. If $c^d_s = 0$ then it means that $y^d_s = y^d_{s-1}$ i.e. topic does not change between the segments. However, when $c^d_s = 1$, then y^d_s is drawn from a Multinomial distribution parameterized by τ. Computation of $P(y^d_s | c^d_s, \tau, y^d_{s-1})$ is done based on two conditions i.e. $\rho(y^d_s, y^d_{s-1})$ when $c^d_s = 0$ or sampling from **Multinomial** (τ) when $c^d_s = 1$.

The segment distribution $P(y_s^d|c_s^d, \tau, y_{s-1}^d)$ is not properly defined for the first segment of every document. Therefore, $c_s^d = 1$ is defined for the first segment which is drawn from Multinomial(τ). Similarly we assume that x_{s1}^d is observed and only unigram is allowed at the beginning of every segment.

4.3 Posterior Inference

The inference problem is related to computing the posterior probability of the hidden variables when the input parameters $\beta, \gamma, \delta, \rho, \Omega$ and the observed variable $\mathbf{w}$ are given. Also, an estimate of the α hyperparameter has to be made. It can be shown that computing the exact inference in our model is intractable. Hence, we need to resort to approximation techniques such as Gibbs sampling [40]. Adoption of Bayesian methods results in some hidden parameters being integrated out instead of being explicitly estimated. This process with Gibbs sampling method is called collapsed Gibbs sampling. Algorithm 1 depicts the collapsed Gibbs sampling used in our approximate inference.

The target distribution is the posterior distribution of the word-topics, the segment-topics, the topic switching variables of the segments, and the bigram status variables. When we use collapsed Gibbs sampling technique, in each iteration, we sample from the conditional distribution of the word-topics in a document conditioned on the topic assignments for all other words except the current word (Line 12 in Algorithm 1). In addition, we also sample the bigram status variable (Line 13). We then sample from the conditional distribution of a segment-topic for a segment and also the corresponding switching variable given the topic assignments for all other words in the current segment (Line 20).

In each iteration of the Gibbs sampling procedure, we only sample a subset of the variables which are directly related to the conditional probability and collapse

Input : $\gamma, \delta, L, K, \rho, \Omega, \beta, Corpus, MaxIteration$
Output: Generation of n-gram words in a topic with the same topic assignments and segments in documents, an estimate of the α hyperparameter

Initialization: Randomly initialize the n-gram word-topic assignments for all the words and segment-topic assignments, topic switch variable for all the segments and the bigram status variable for all the words;
Zero all count variables such as m_{wv}, p_{lwt}, n_{lv}, n_{l^d}, n_{d_0}, and n_{d_1};
Compute P_{d_k} for all values of $k \in \{1, \cdots, K\}$ and all documents;
Compute n_{lv} and p_{lwt} for all values of $l \in \{1, \cdots, L\}$ and all words;
Compute n_{l^d} for all values of $l \in \{1, \cdots, L\}$ and all documents and their segments;
if *performing parameter value estimation* **then**
    Initialize α using Equations 4.5, 4.6, 4.8, 4.7, 4.9;
end
Randomize the order of the documents in the corpus;
for $iter \leftarrow 1$ **to** $MaxIteration$ **do**
    foreach *word i according to order* **do**
        Exclude word i and its assigned topic l from variables n_{l^d} and n_{li};
        $(newl, newx) \leftarrow$ sample new word-topic for word i and bigram switching variable using Equation 4.3;
        if *(newx==1)* **then**
            Assign *newl* as the new word-topic;
        end
        Update variables n_{l^d}, n_{li}, p_{lit} and m_{iv} using the new word-topic *newl* for word i;
        if *Entered a new segment j* **then**
            Exclude segment j and its assigned topic k from variable P_{d_k};
            $(newk, newc) \leftarrow$ sample new segment-topic and segment switching variable for segment j using Equation 4.4;
            if $newc == 1$ **then**
                Assign *newk* as the new segment-topic for segment j;
            end
            Update variable P_{d_k} using the new segment-topic *newk* for segment j and also n_{d_c};
            if *performing parameter value estimation* **then**
                Update α using Equations 4.5, 4.6, 4.8, 4.7, 4.9;
            end
        end
    end
end
Compute posterior estimates using Equations 4.10,4.11,4.12,4.13,4.14,4.15;

Algorithm 1: Inference algorithm for `NTSeg`.

out the nuisance variables. We perform this step repeatedly until we arrive at some approximation. A variable is sampled from the conditional distribution given that the assignments for all other variables are known which is a standard procedure in a Gibbs sampler. As the list of words is being scanned along with the bigram status variables, the sampler keeps track of any new segment being encountered. For each new segment, the sampler decides about the topic assignment of the segment i.e., whether it should assign the current segment to the same topic as the previous segment or a new segment-topic. If the segment has to be assigned to a new segment-topic, the sampler estimates the probability of assigning the segment to the segment-topic. These probabilities are computed from the conditional distribution for a segment given all other topic assignments to every other segment and all words in the segment as depicted in Algorithm 1 from Lines 10 to 25.

We need to compute the two conditional distributions:

$$P(z^d_{si}, x^d_{si} | z^d_{\neg si}, x^d_{\neg si}, \mathbf{w}, \mathbf{c}, \mathbf{y}, \mathbf{x}, \alpha, \beta, \gamma, \delta, \rho, \Omega) \tag{4.1}$$

$$P(y_s, c_s | \mathbf{z}, y^d_{\neg s}, c^d_{\neg s}, \mathbf{w}, \mathbf{x}, \alpha, \beta, \gamma, \delta, \rho, \Omega) \tag{4.2}$$

Note that $w^d_{\neg si}$ defines all the words in the segment except the current word w^d_{si}. $z^d_{\neg si}$ is the word-topic assignments for all other words except the current word w^d_{si}. Beginning with the joint probability of a dataset, and using the chain rule, we obtain the conditional probabilities conveniently. We obtain the following equations:

$$P(z^d_{si}, x^d_{si} | \mathbf{w}, z^d_{\neg si}, x^d_{\neg si}, \mathbf{y}, \mathbf{c}, \alpha, \beta, \gamma, \delta, \rho, \Omega) \propto$$
$$(\alpha_{y_s z^d_{si}} + n_{z^d_{si}} - 1) \times (\gamma_{x^d_{si}} + p_{z^d_{s(i-1)} w^d_{s(i-1)} x^d_{si}} - 1)$$
$$\times \begin{cases} \frac{\beta_{w^d_{si}} + n_{z^d_{si} w^d_{si}} - 1}{\sum_{v=1}^{W} \left(\beta_v + n_{z^d_{si} v}\right) - 1} & \text{if } x^d_{si} = 0 \\ \frac{\delta_{w^d_{si}} + m_{w^d_{si} w^d_{si-1}} - 1}{\sum_{v=1}^{W} \left(\delta_v + m_{w^d_{si-1} v}\right) - 1} & \text{if } x^d_{si} = 1 \end{cases} \tag{4.3}$$

$$P(y_s, c_s | \mathbf{z}, y^d_{\neg s}, c^d_{\neg s}, \mathbf{w}, \mathbf{x}, \alpha, \beta, \gamma, \delta, \rho, \Omega) \propto$$
$$\begin{cases} (\rho_{y^d_s} + P_{d_{y^d_s}} - 1) \times \left(\frac{\prod_{l=1}^{L} \prod_{j=0}^{n^d_{sl}-1} (\alpha_{y^d_s l} + j)}{\prod_{j=0}^{N^d_s - 1} \left(\sum_{l=1}^{L} \alpha_{y^d_s l} + j \right)} \right) \left(\frac{n_{d_1} + \Omega}{N^d_s + 2\Omega} \right) & \text{if } c_s = 1 \\ \left(\frac{\prod_{l=1}^{L} \prod_{j=0}^{n^d_{sl}-1} (\alpha_{y^d_s l} + j)}{\prod_{j=0}^{N^d_s - 1} \left(\sum_{l=1}^{L} \alpha_{y^d_s l} + j \right)} \right) \left(\frac{n_{d_0} + \Omega}{N^d_s + 2\Omega} \right) & \text{if } c_s = 0 \;\&\; s > 1 \;\&\; y_s = y_{(s-1)} \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

In Equations 4.3 and 4.4, n_{zw} is the number of times that the word w is assigned to the word-topic z as a unigram. For example in Equation 4.3, when we write $n_{z^d_{si} w^d_{si}}$ then it denotes the total number of times a word w^d_{si} has been assigned to word-topic z^d_{si}. m_{wv} is the number of times the word v is assigned as the second word of a bigram given the previous word w given the same topic of the previous word (the number of times z^d_{si}) appears after seeing the bigram switch variable x^d_{si} on word w^d_{si-1} and w^d_{si} conditioned on the topic z^d_{si-1} of the previous word w^d_{si-1}. p_{zwt} denotes the number of times the status variable $x = t$ (0 or 1) in the same topic z as the the previous word w. y^d_s is the segment-topic that has been assigned to the paragraph s in document d. $n_{z^d_{si}}$ is the number of times a word in segment s of document d is assigned to

word-topic z. n_{d_0} and n_{d_1} is the number of times the switching variable c_s is set of 0 and 1 in the document d respectively. y_s is the segment-topic assignment for segment s in the document d. $P_{d_{y_s^d}}$ is the number of times a segment in the document d has been assigned to the segment-topic y_s^d. When we write P_{d_k}, it means the number of times a segment in the document d has been assigned to the segment-topic k. $y_{\neg s}^d$ is the segment-topic assignments for all the segments except the current segment s. $\alpha_{y_s z_{si}}$ is the z_{si}^{th} component in α_{y_s}. We will present the complete posterior inference derivation in Appendix A.

Note that in our model the hyperparameter α captures the relationships between the segment-topics and word-topics. This hyperparameter must be estimated from the data. Although there are many ways to estimate this hyperparameter [231], we adopt moment matching which is computationally less expensive [231], [161]. Therefore in each iteration of the collapsed Gibbs sampling (Line 25 of Algorithm 1), we update:

$$\bar{m}_{kl} = \frac{1}{\hat{n}_k} \sum_{\hat{s} \in \hat{S}_k} \frac{n_{z_{si}^d}}{N_s^d} \tag{4.5}$$

$$\bar{v}_{kl} = \frac{1}{\hat{n}_k} \sum_{\hat{s} \in \hat{S}_k} \left(\frac{n_{z_{si}^d}}{N_s^d} - \bar{m}_{kl} \right)^2 \tag{4.6}$$

$$\alpha_{kl} \propto \bar{m}_{kl} \tag{4.7}$$

$$m_{kl} = \frac{\bar{m}_{kl}(1 - \bar{m}_{kl})}{\hat{v}_{kl}} - 1 \tag{4.8}$$

$$\sum_{l=1}^{L} \alpha_{kl} = \exp\left(\frac{\sum_{l=1}^{L} \log(m_{kl})}{L - 1} \right) \tag{4.9}$$

where $\hat{S}_k$ is the set of segments assigned to the segment-topics k. $\hat{n}_k$ is the number of segments assigned to the segment-topic k. $\bar{m}_{kl}$ and $\bar{v}_{kl}$ are the sample mean and sample variance, respectively, which is computed over all the segments assigned to the segment-topic k.

The posterior estimates for θ, ϕ, ψ, π, τ, σ are:

$$\hat{\theta}_z^{(s)} = \frac{\alpha_{yz} + n_{sz}^d}{\sum_{l=1}^{L}(\alpha_{yl} + n_{sl}^d)} \quad (4.10) \qquad \hat{\phi}_{zw} = \frac{\beta_w + n_{zw}}{\sum_{v=1}^{W}(\beta_v + n_{zv})} \quad (4.11)$$

$$\hat{\psi}_{zwt} = \frac{\gamma_t + p_{zwt}}{\sum_{t=0}^{1}(\gamma_t + p_{zwt})} \quad (4.12) \qquad \hat{\sigma}_{wv} = \frac{\delta_w + m_{wv}}{\sum_{v=1}^{W}(\delta_v + m_{wv})} \quad (4.13)$$

$$\hat{\pi}_c = \frac{\Omega_c + n_{d_c}}{\sum_{c=0}^{1}(\Omega_c + n_{d_c})} \quad (4.14) \qquad \hat{\tau}_y = \frac{P_{d_y} + \rho_y}{\sum_{k=1}^{K}(P_{d_k} + \rho_k)} \quad (4.15)$$

4.4 Experiments and Results

Evaluation of topic models is a challenging task [253]. Simply showing the highly probable n-gram words obtained from each topic may not be able to portray the underlying strengths or weaknesses of a topic model. Therefore, we evaluate our model on several text mining tasks including the ability to support fine grained topics with n-gram words in the correlation graph, the ability to segment a document into topically coherent sections, document classification, and document likelihood estimation.

In each experiment, we chose several existing closely related comparative methods for comparison purpose. We will describe those comparative methods in the subsections that follow. For our proposed framework, NTSeg, the segment granularity is basically a paragraph because topical changes typically occur at paragraph boundary and this strategy is also used in [231]. Note that NTSeg can also work at the granularity of a sentence which has also been used in one of our experiments (refer Section 4.4.2). In our experiments, the number of iterations for the Gibbs sampler is 1000 which is the value of the $MaxIteration$ used in Algorithm 1. We have chosen the following hyperparameter values $\beta = 0.01$, $\gamma = 0.1$, $\delta = 0.1$, $\Omega = 0.1$, and $\rho = 0.1$. Other topic models such as TNG, LDSEG etc, also assume fixed hyper-

parameter values. We did not perform any stemming, but removed stopwords[1] from the collection.

4.4.1 Correlation Graph

NTSeg produces two levels of topics, namely, segment-topics and word-topics. A word-topic is comprised of n-grams. We show the correlation graph for the purpose of depicting how our model finds correlations among various segment-topics and word-topics. We only show a part of the correlation graph in Figures 4.7 and 4.8. We present words in a word-topic in a box with some high probable n-grams. For each document-topic, we rank the word topic according to the Dirichlet parameters. We mainly follow the details outlined in the following two works [161], [231], which go like this: For each word-topic z in the graph, we have a box where the word-topics is represented by the most probable words. For each document-topic k, we rank the word-topics z according to the Dirichlet parameter α_{kl}.

We have used the OHSUMED[2] and NIPS document collections to show the correlation graph. The collection is composed of 348,566 documents with 154,711 words in the vocabulary without stopwords. Our intention is to also show that our model can scale to large document collections. In comparison to the OHSUMED collection, the NIPS collection is significantly small.

We have experimented by varying both number of the word-topics L and the number of the segment-topics K. L was varied from 50 to 200 in steps of 50 whereas K was varied from 50 to 150 in steps of 50. However, we did not observe significant difference in the quality of the results. The resulting correlation graphs are shown in Figures 4.7 and 4.8 which is obtained by setting $L = 200$ and $K = 100$. We show the graph obtained from our NTSeg model. Note that other models such as

[1] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop
[2] http://ir.ohsu.edu/ohsumed/ohsumed.html

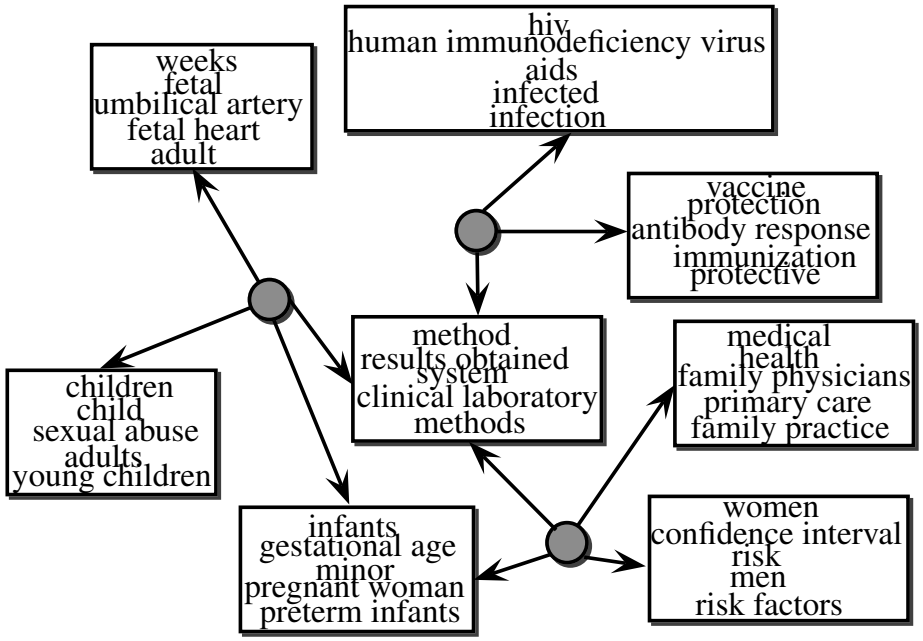


Figure 4.7: Correlation identified by NTSeg between the word-topics and the segment-topics on OHSUMED collection considering $L = 200$ and $K = 100$. Each circle shows a segment-topic and each box corresponds to a word-topic. We can notice that a segment-topic can capture correlations between several word-topics.

PAM, LDCC, LDSEG, GD-LDA [35] and CTM [15], only form unigrams in a topic leading to ambiguous interpretation. For example, presenting the unigram "confidence" will not be that insightful in a correlation graph. In contrast, presenting the term "confidence interval" is more meaningful as shown in Figure 4.7. We also show the correlation graph obtained from the previously proposed GD-LDA [35] model, where we only notice unigram words in the correlation graph.

4.4.2 Topic Segmentation Experiment

The purpose of this experiment is to show how well NTSeg generates segmentation of documents corresponding to coherent topical units. The segmentation information is obtained via the segmentation switch variable c_s^d which gives the segment topic change-points in the document. In our problem setting we know the segment boundaries in advance such as paragraphs or sentences, but we do not know the word and

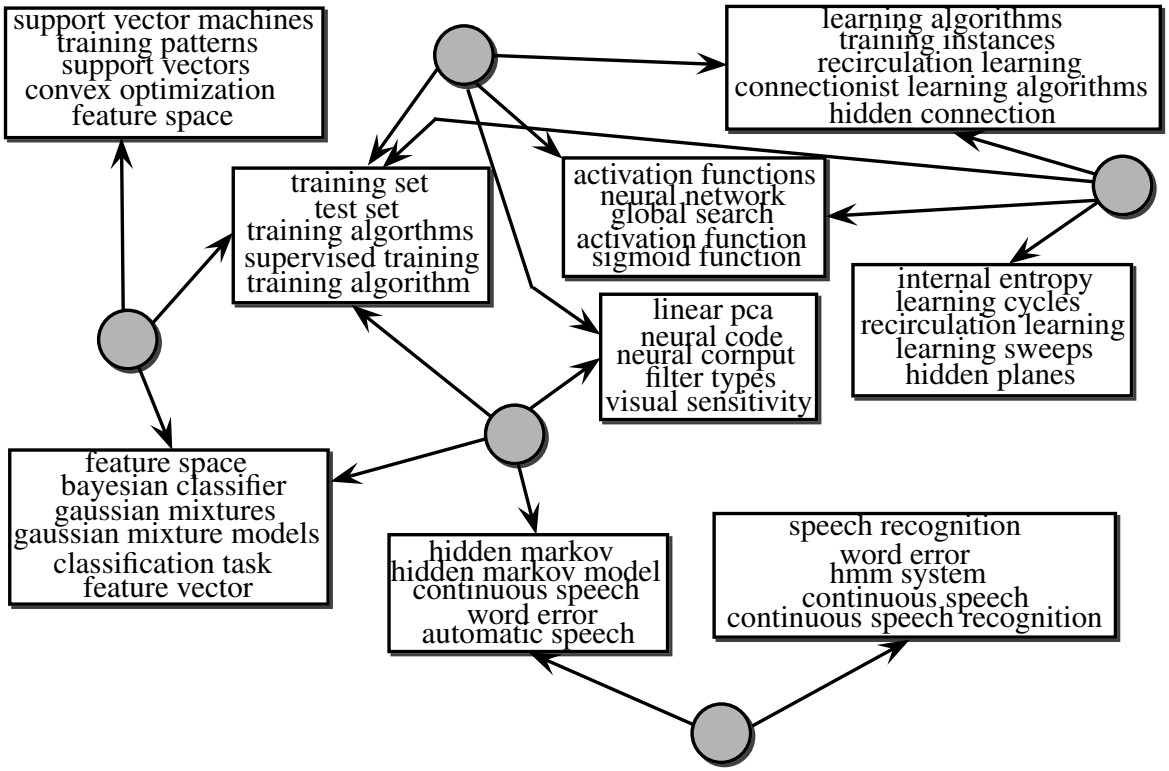


Figure 4.8: Correlation identified by our model between the word-topics and the document-topics. Each circle shows a document-topic and each box corresponds to a word-topic. We can notice that a document-topic can capture correlations between several word-topics.

segment topics. Our purpose is thus to learn the segment and word topics from the document collection. The prediction output of the segment status variable will define the segmentation of a document. To evaluate the performance, we make use of the annotated segmentation information. We use two standard metrics, namely, Pk and WinDiff which are widely used in the topic segmentation literature [230]. As described in [230], Pk is defined as the probability that two segments drawn randomly from a document are incorrectly identified as belonging to the same topic [10]. WinDiff [199] moves a sliding window across the text and counts the number of times the hypothesized and reference segment boundaries are different from within the window. The lower the values obtained for these two metrics, the better is the segmentation result.

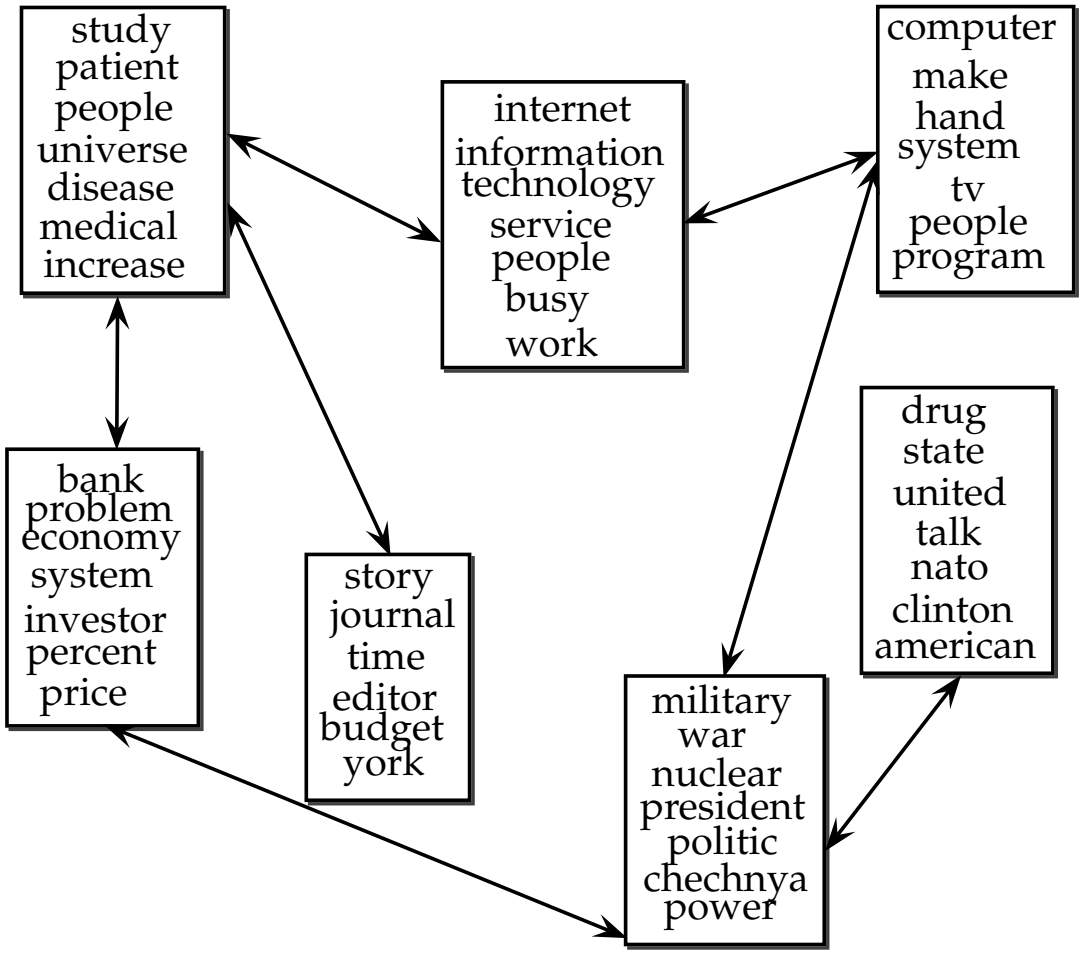


Figure 4.9: Correlation identified by the CTM model between the word-topics and the document-topics. We can see that that unigrams are generated by the model which at times are not very insightful to a reader.

We use two publicly available datasets that contain segment boundaries corresponding to the topic changes. The first dataset, called Lectures in our experiment, consists of spoken lecture transcripts from an undergraduate physics class and a graduate artificial intelligence class. The transcripts consist of a 90 minute lecture recording and have 500 to 700 sentences with about 9000 words. Note that here the segment granularity is a sentence. More details about this dataset can be obtained from [230]. Our second dataset, called Books in our experiment, is the books[3] dataset in which each document is a chapter extracted from a medical textbook.

[3]http://groups.csail.mit.edu/rbg/code/bayesseg/

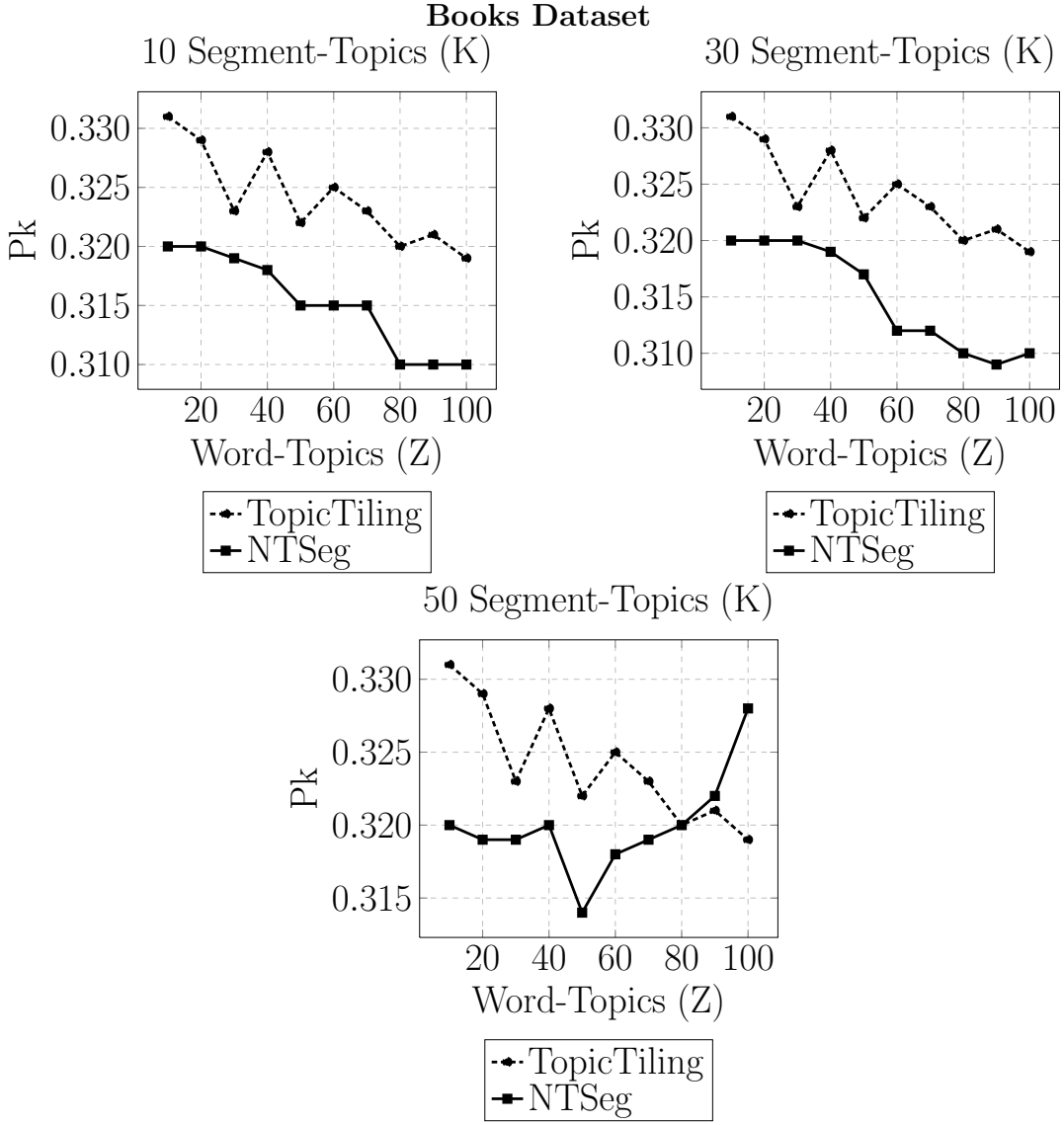


Figure 4.10: Topic segmentation results of our model compared with the `TopicTiling` model in terms of Pk metric.

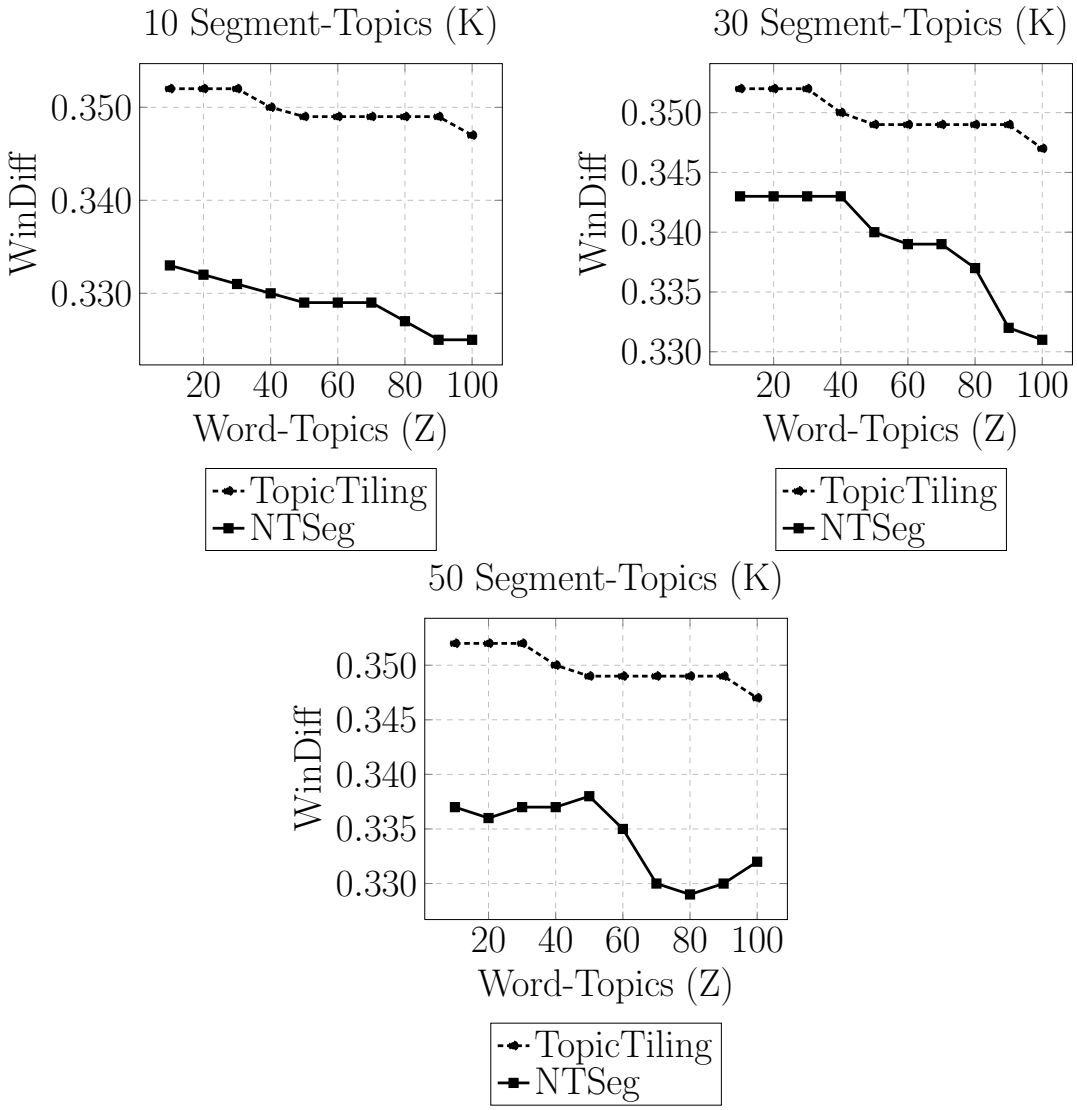


Figure 4.11: Topic segmentation results of our model compared with the `TopicTiling` model in terms of $WinDiff$ metric.

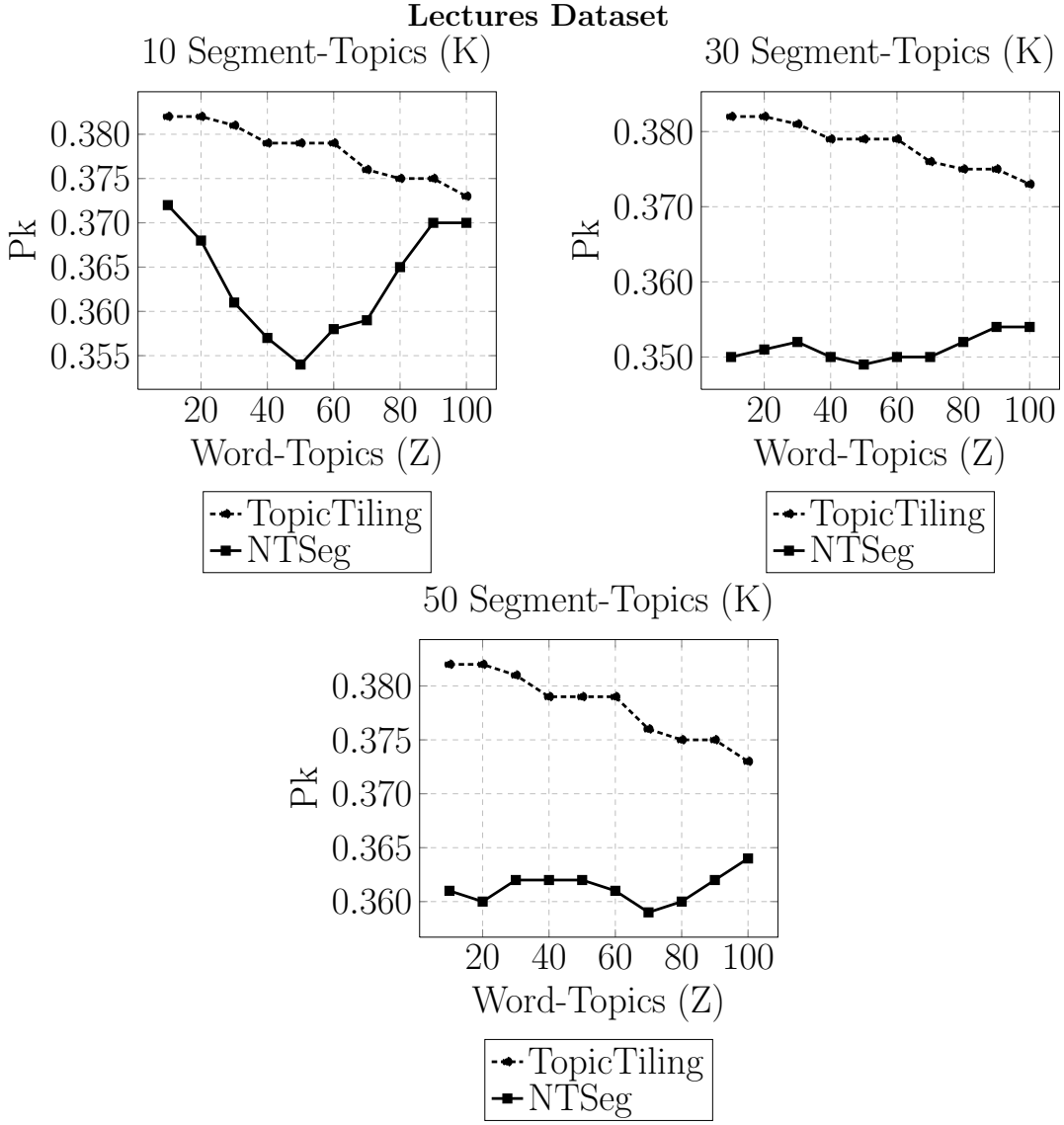


Figure 4.12: Topic segmentation results of our model compared with the `TopicTiling` model in terms of Pk metric.

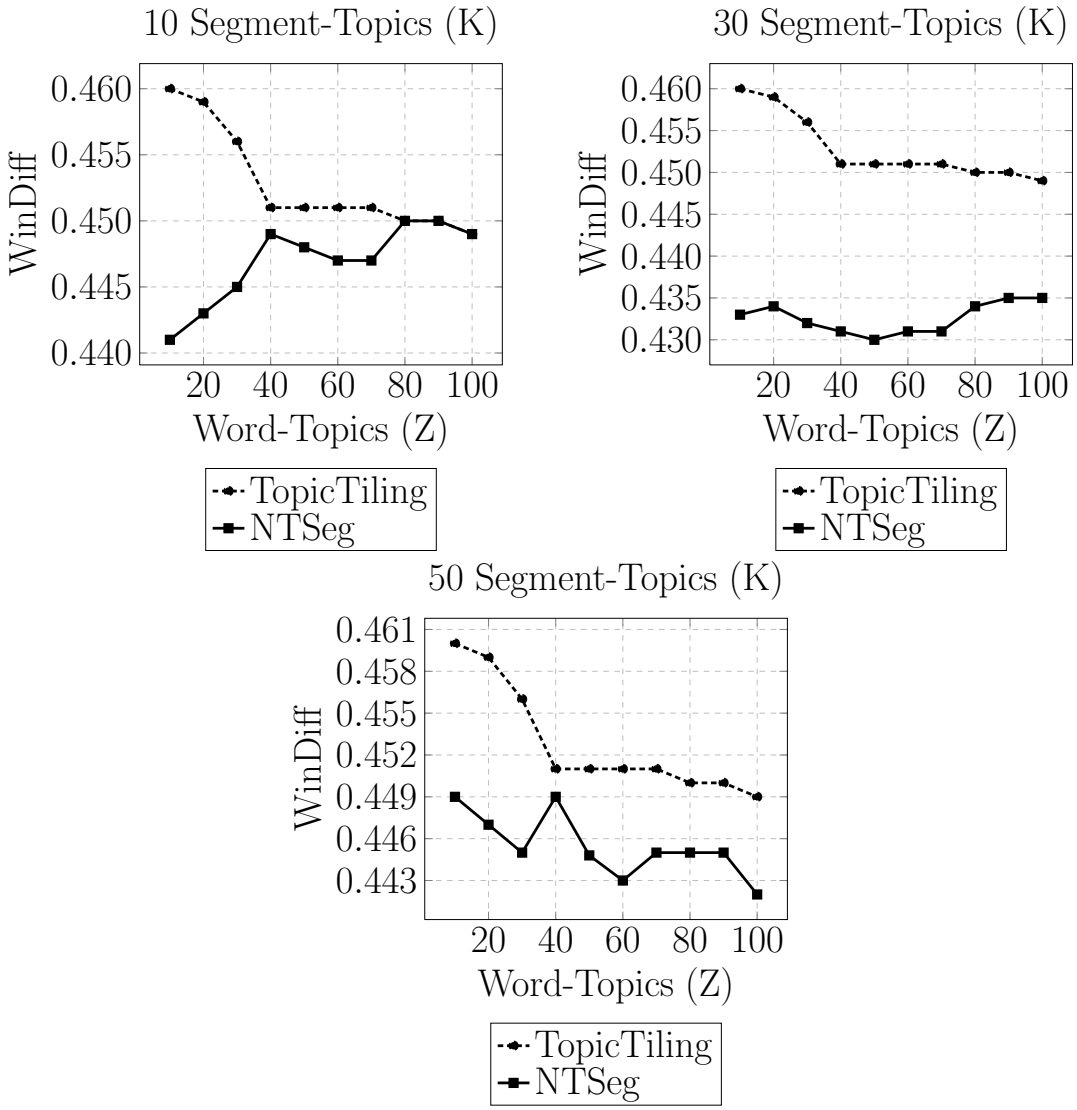


Figure 4.13: Topic segmentation results of our model compared with the `TopicTiling` model in terms of $WinDiff$ metric.

	Precision	Recall	F-Measure
LDSEG	0.580	0.420	0.487
PAM	0.550	0.450	0.495
LDACOL	0.400	0.300	0.343
TNG	0.490	0.420	0.452
PDLDA	0.580	0.500	0.537
NTSeg	**0.640**	**0.520**	**0.574**

Table 4.1: Document classification results for the Computer Dataset of the 20 Newsgroups corpus.

	Precision	Recall	F-Measure
LDSEG	0.440	0.400	0.419
PAM	0.500	0.330	0.398
LDACOL	0.420	0.370	0.393
TNG	0.560	0.470	0.511
PDLDA	0.580	0.510	0.543
NTSeg	**0.620**	**0.560**	**0.588**

Table 4.2: Document classification results for the Science Dataset of the 20 Newsgroups corpus.

We chose a recently proposed topic segmentation method `TopicTiling` [220] which has outperformed many state-of-the-art text segmentation models proposed in the literature and chose the best performing variant of `TopicTiling` from [220]. Note that `TopicTiling` only has the notion of word-topics. For each of the segment and word-topics, we run the Gibbs sampler five times and take the average of the Pk and WinDiff values at the end of the fifth run.

We illustrate the segmentation results in Figures 4.10, 4.11, 4.12, and 4.13. From the results, we note that our model performs extremely well in both datasets compared to the state-of-the-art topic segmentation model. Using a two-tailed significance test, our results are statistically significant with $p < 0.05$ against `TopicTiling`. In the Books dataset, `NTSeg` performs reasonably better, but the improvement obtained is not very high considering both Pk and WinDiff metrics. However, good improvement is obtained in the Lectures dataset using both metrics.

	Precision	Recall	F-Measure
LDSEG	0.390	0.320	0.352
PAM	0.540	0.490	0.514
LDACOL	0.550	0.410	0.470
TNG	0.550	0.450	0.495
PDLDA	0.590	0.410	0.484
NTSeg	**0.620**	**0.570**	**0.594**

Table 4.3: Document classification results for the Politics Dataset of the 20 Newsgroups corpus.

	Precision	Recall	F-Measure
LDSEG	0.330	0.320	0.325
PAM	0.368	0.360	0.363
LDACOL	0.200	0.180	0.189
TNG	0.340	0.290	0.313
PDLDA	0.380	0.210	0.271
NTSeg	**0.420**	**0.380**	**0.399**

Table 4.4: Document classification results for the Sports Dataset of the 20 Newsgroups corpus.

4.4.3 Document Classification Experiment

We conduct document classification experiment using topic models. In the training phase, a topic model is learned for each class using the set of training documents in that class. In testing, to conduct document classification for a testing document, we compute the likelihood of the testing document against each trained topic model for each class. The testing document is classified to the model that produces the highest likelihood. Note that this procedure is also used in [161].

We measure the classification performance using precision, recall and F-measure. The meaning of precision for a class is the number of true positives divided by the total number of documents predicted to that class. Recall is defined as the number of true positives divided by the total number of elements that actually belong to that class in the gold standard. F-measure is the harmonic mean of precision and recall.

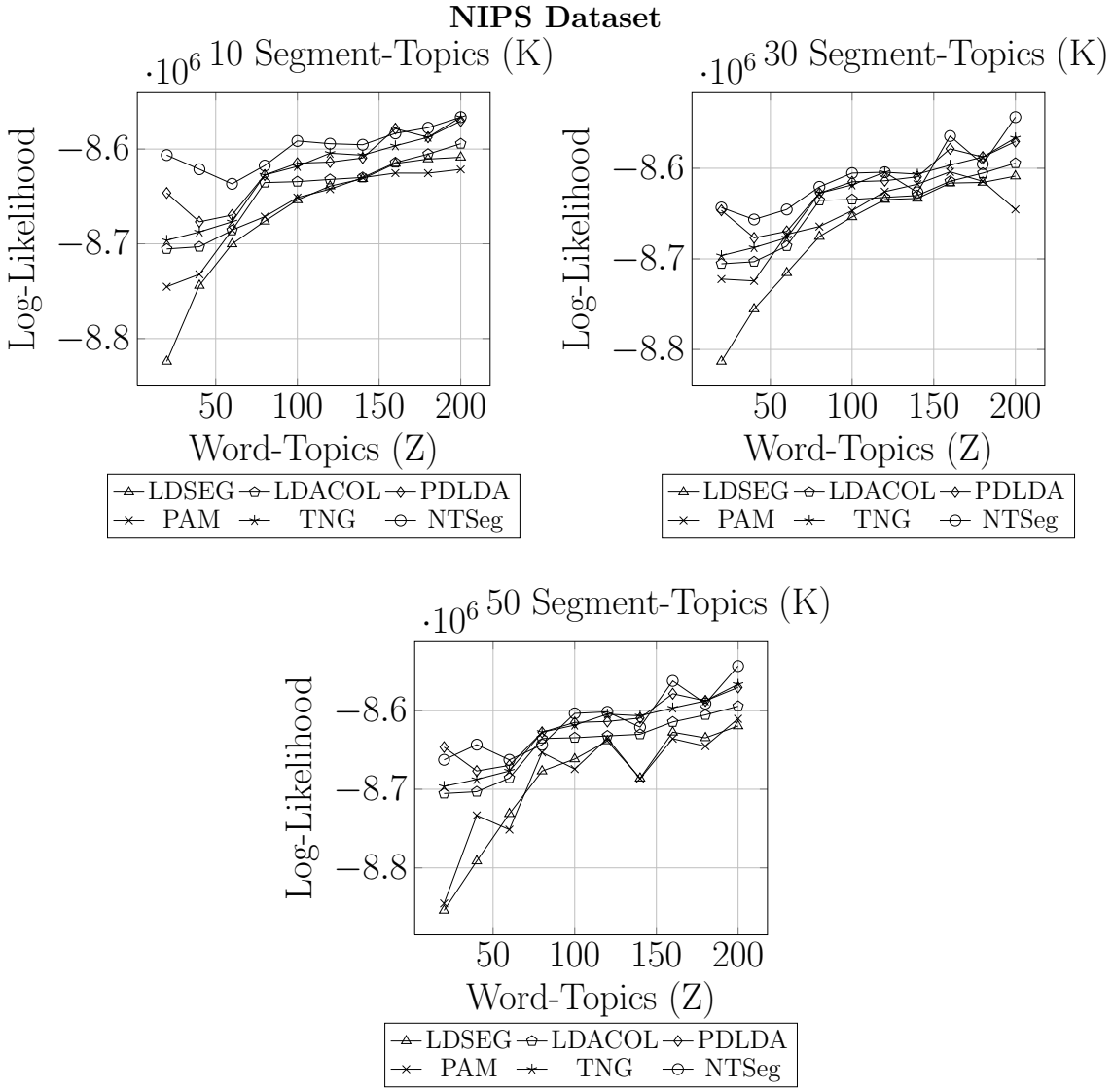


Figure 4.14: Document modeling results of our model compared to other topic models.

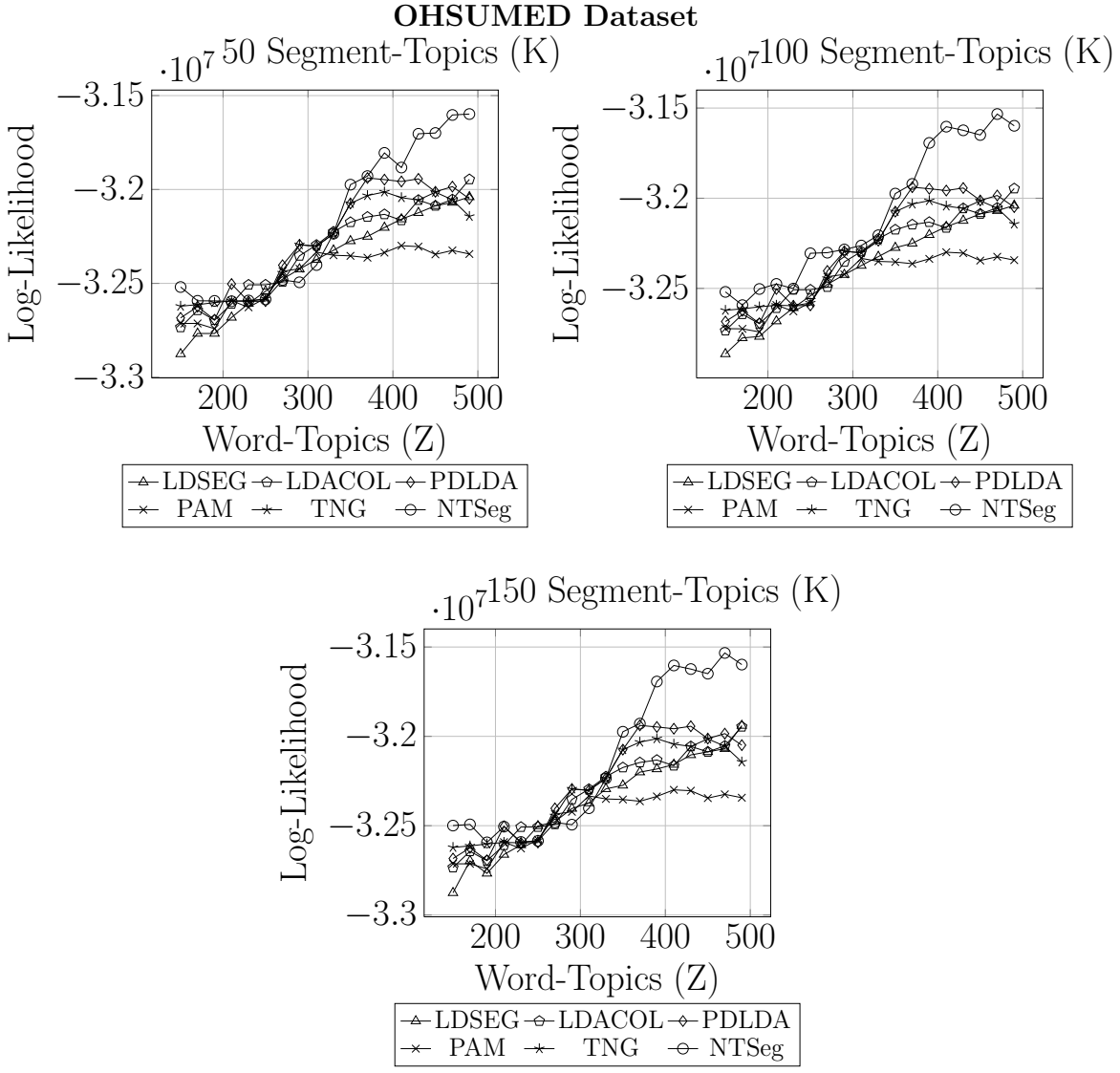


Figure 4.15: Document modeling results of our model compared to other topic models.

We use the 20 Newsgroups corpus[4] and generated four datasets. The first dataset comprises of documents related to computer technology (the "comp" directory in the dataset). It is composed of several classes such as "graphics", "windows", "hardware", etc. Each of these classes consists of 1000 documents. We split the documents in each of these classes into 75% training and 25% test documents. For each class, we trained and tested the model by varying the number of word-topics from 10 to 100 in steps of 10 and the number of segment-topics from 10 to 50 in steps 20. We compute precision and recall using the test set for each class for each word-topic and segment-topic values and then we compute the average result for one class across all word-topics and segment-topics. Similarly, we follow the same precision and recall computation for all classes. Finally we compute the average over all precision and recall values for all the classes. We then compute F-measure from the obtained precision and recall values. The experimental setup is similar for the other three datasets, namely, "sci" (called Science Dataset), "politics" (called Politics Dataset), and "sports" (called Sports Dataset).

The comparative methods include LDSEG, PAM, LDACOL, TNG, and PDLDA. All these models are described in Chapter 2. Note that some of the comparative methods such as TNG, PDLDA, and LDACOL have no notion of segment-topics.

The classification performance results are presented in Tables 4.1, 4.2, 4.3 and 4.4. We can observe that in all the datasets our model, NTSeg, has outperformed all the comparative methods. Compared to all the comparative methods, our results are also statistically significant using the sign test with $p < 0.05$. Gain obtained in the Computer and Science datasets is more when compared to the gain in Sports and Politics datasets. PDLDA also proved to be a better model in comparison to the other comparative methods.

[4]http://qwone.com/~jason/20Newsgroups/

4.4.4 Document Likelihood Experiment

Another evaluation scheme to compare the relative performance of topic models is to study how the models generalize on an unseen data. The entire corpus in this method is first split into training and testing set. The training set generally contains more number of documents as compared to the testing set. A model is first learned on the training data, and the testing set is used to measure the generalization performance of the topic models. In the topic modeling literature, metrics such as perplexity computation or log-likelihood have often been used. For example, PAM uses empirical log-likelihood [60] as an evaluation metric and so does a recently proposed method GD-LDA [35]. Log-likelihood has also been widely used as one of the evaluation metrics, for example in [15]. We chose log-likelihood metric for comparing the topic models. The comparative methods here are LDSEG, PAM, LDACOL, TNG, and PDLDA.

We use the NIPS dataset[5]. The NIPS collection is widely used in the topic modeling literature. Note that the original raw NIPS dataset consists of 17 years of conference papers. But we supplemented this dataset by including some new raw NIPS documents[6] and it has 19 years of papers in total. Our NIPS collection consists of 2741 documents comprising of 453,606,9 non-unique words and 94961 words in the vocabulary. In addition to the NIPS collection we also use the OHSUMED collection.

In order to calculate the likelihood of held-out data, we must integrate out the sampled multinomials and sum over all possible topic assignments which has no closed-form solution. Griffiths et al. [86] have used Gibbs sampling for computing such approximations. First, we randomly split each of the datasets into 80% training and 20% testing. We trained each of the topic models on the training set. We then tested the models on the testing set by running the inference algorithms five times for each word-topic and segment-topic pair. We then took average value for all five

[5]http://www.cs.nyu.edu/~roweis/data.html
[6]http://ai.stanford.edu/~gal/Data/NIPS/

runs. We varied the number of segment-topics from 10 to 50 in steps of 20 and the number of word-topics from 20 to 200 in steps of 20 in the NIPS collection. As the OHSUMED collection is larger compared with the NIPS collection, so we varied the number of segment-topics from 50 to 150 in steps of 50 and word-topics from 150 to 490 in steps of 20.

From the results in Figures 4.14 and 4.15 we can see that in the NIPS collection, NTSeg performs better than the comparative methods especially when the number of segment-topics is 10. However, its performance deteriorates a bit when the number of segment-topics is increased, but still remains competitive with the comparative methods. Moreover, we notice that as the number of word-topics increases, the performance of NTSeg deteriorates to some extent in the NIPS collection. However, in the OHSUMED collection, NTSeg again performs better against the comparative methods when the number of word-topics is increased. We can observe that NTSeg outperforms the comparative methods considerably when the number of segment-topics is 100. The results suggest that NTSeg can perform very well on large document collections as large collections provide richer information about word co-occurrences.

4.5 Closing Remarks

We have presented a generative topic discovery model in this chapter, known as NTSeg, which maintains the document's structure such as paragraphs and sentences and also keeps the order of the words in the document intact. NTSeg incorporates the notion of word-topics and segment-topics. We have conducted extensive experiments and shown results using both qualitative analysis where we show the n-gram words in the correlation graph and quantitative performance. Experimental results demonstrate that by relaxing the bag-of-words assumption in each segment improves the performance of the model.

Chapter Five

Modeling Temporal Dynamics in Text Documents

Chapter Summary

This chapter presents a topic model that captures the temporal dynamics in the text data along with topical phrases. Previous approaches have relied upon bag-of-words assumption to model such property in a corpus. This has resulted in an inferior performance with less interpretable topics. Our topic model can not only capture changes in the way a topic structure changes over time but also maintains important contextual information in the text data. Finding topical n-grams, when possible based on context, instead of always presenting unigrams in topics does away with many ambiguities that individual words may carry. We derive a collapsed Gibbs sampler for posterior inference.

5.1 The Case for Capturing N-grams over Time

Popular text processing models such as LDA [23] and TOT model [260] assume that the order of words in a document is not important. As a result, these models lose important collocation information in documents. For example, LDA, due to its bag-of-words assumption, fails to capture a phrase such as "acquired immune deficiency syndrome" which is one of model's shortcoming. Also, if one uses the TOT model on the NIPS document collection, then word such as "networks" in a topic will not convey much insight to a human being, instead presenting "neural networks" seems to be more insightful. Thus by presenting words along with their context in a topic can help a person obtain better insights about a word in a topic.

Data is ever evolving and so are topics. At one point in time, one topic may be highly popular than others but this popularity may eventually decline. In Figure 5.1, we present an example of such topical changes over time. For example, in the year 2010, "Burj Khalifa" in the United Arab Emirates was the most dominant topic all over the world. Then people stopped discussing about it after some period of

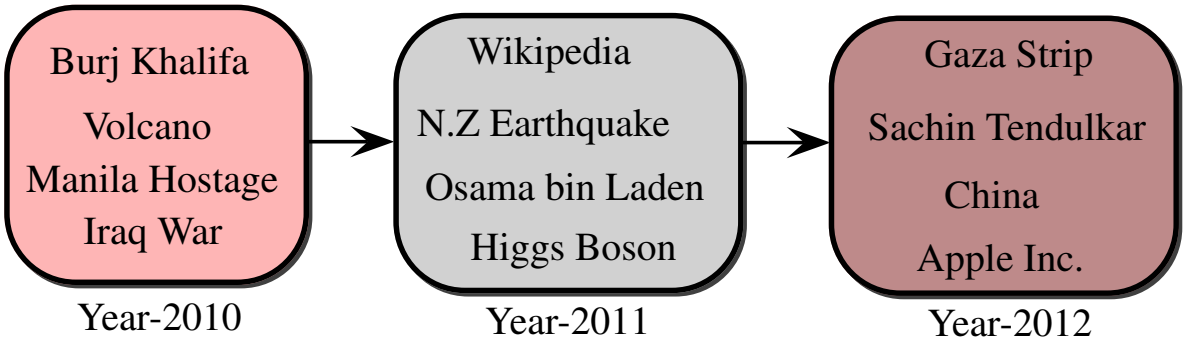


Figure 5.1: An illustration about how some entities come into existence and fade over time. They are then taken over by other entities. It means that topics tend to change over time, and what remains dominant today might not be dominant after some period of time.

time, which was then taken over by topics such as an "Earthquake in New Zealand". Models such as LDA cannot capture such time dependent changes in topics. In order to capture such structure in the data many models have been proposed, for example, [18], [256], [133] and TOT model is also one among them. The model incorporates time along with the word co-occurrence patterns. A limitation of this model and other related models is that they fail to capture n-gram words or phrases in a topic. This in turn results in less coherent words in each topic and topics tend to become less interpretable over time. A common limitation of the n-gram topic models, such as [252], [117], [166], etc, is that they cannot capture how topics evolve over time.

We present a model which can not only consider the local contextual information inherent in the document, but also captures the way in which the topic structure changes over time. By maintaining the word order in the document and capturing phrases in topics can help us find words in topics which convey better meaning to the reader. In our model, a continuous distribution over time is associated with each topic. Topics generate words and observed time-stamp values. The model automatically determines whether to form a unigram or combine with the previous word in each time-stamped document.

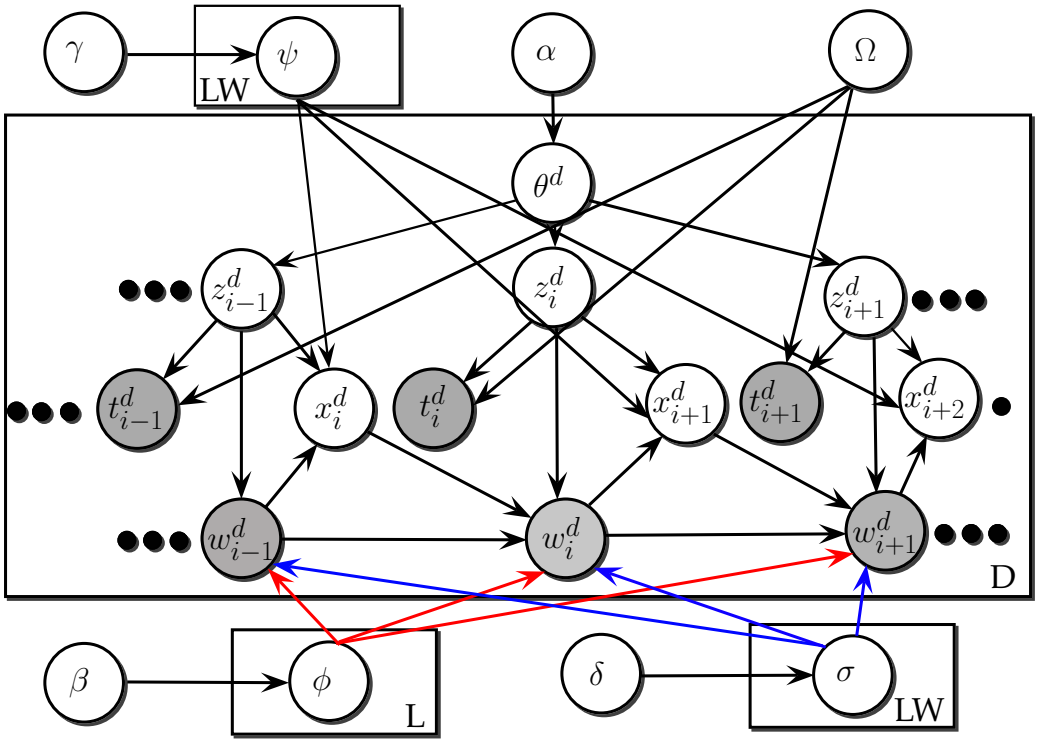


Figure 5.2: Graphical model of our proposed n-gram topics over time model.

5.2 Our N-gram Temporal Topic Model

The graphical model is shown in Figure 5.2, where δ is the Dirichlet prior of σ and σ is the Discrete distribution. Our model is not just a simple extension of the TOT model because it allows us to find topical phrases over time which is not possible using a topic model relying on a simple bag-of-words assumption. The input to our model is the entire document with word order kept intact, rather than the traditional term-document matrix. Our model consists of a bigram switch/status variable $\mathbf{x}$ which keeps track whether a word forms a bigram with the preceding word over time. If it is possible to form a bigram then x_i^d is set to 1 else $x_i^d = 0$. Combining successive n-grams in sequence gives rise to higher order n-grams ($n > 2$) or phrases. γ in Figure 5.2 is the Dirichlet prior of ψ where ψ is the Bernoulli distribution of the status variable x_i^d with respect to the previous word where x_i^d is the bigram switch variable between w_{i-1}^d and w_i^d in document d. We assume a hypothetical unigram w_h^d at the beginning of every document. Also, we assume that the first bigram

switch variable x_1^d is observed and we allow only a unigram at the beginning of the document. The TNG and the LDACOL models can also capture topical phrases in topics by introducing a bigram status variable, but they cannot capture temporal information. We assume a continuous distribution over time associated with each topic and find patterns which are localized over time in the corpus. The reason for adopting a continuous distribution is that it does away with the time discretization process where a major hurdle is the selection of an appropriate time slice. From the graphical model, we can infer that topics are responsible for generating both words and observed time-stamps. Our model not only captures n-gram words in a document but also temporal information. Also note that we assume there is a time-stamp value associated with every word in a document. This time-stamp is basically the time-stamp of the document itself. During model fitting the time-stamp values from the document are copied to the words in that document. Another point to be noted is that the topic allocation in a phrase for two terms may be different. In order to tackle this, we assume the topic assignment for the entire phrase as the topic assigned to the "head noun" in that phrase. This assumption simplifies our model to some extent which speeds up inference algorithm without affecting upon the results considerably. In PDLDA, the authors have relaxed this assumption, but from their graphical model we note that the complexity of their model has rather increased. The generative procedure of our model is shown below.

1. Draw **Discrete**(ϕ_z) from **Dirichlet**(β) for each topic z
2. Draw **Bernoulli**(ψ_{zw}) from **Beta**(γ) for each topic z and each word w
3. Draw **Discrete**(σ_{zw}) from **Dirichlet**(δ) for each topic z and each word w
4. For every document d, draw **Discrete**(θ^d) from **Dirichlet**(α)
    (a) For each word w_i^d in document d
        i. Draw x_i^d from **Bernoulli**($\psi_{z_{i-1}^d w_{i-1}^d}$)

ii. Draw z_i^d from **Discrete**(θ^d)

iii. Draw w_i^d from **Discrete**$(\sigma_{z_i^d w_{i-1}^d})$ if $x_i^d = 1$

iv. Otherwise, Draw w_i^d from **Discrete**$(\phi_{z_i^d})$

v. Draw a time-stamp t_i^d from **Beta**$(\Omega_{z_i^d})$

Input : $\gamma, \delta, \alpha, T, \beta, Corpus, MaxIteration$
Output: Topic assignments for all the n-gram words with temporal information

Initialization: Randomly initialize the n-gram topic assignment for all words;
Zero all count variables;
for *iteration* $\leftarrow$ *1 to MaxIteration* **do**
  for $d \leftarrow$ *1 to* D **do**
    for $w \leftarrow$ *1 to* N_d *according to word order* **do**
      Draw z_w^d, x_w^d defined in Equation 5.1;
      if $x_w^d \leftarrow$ *0* **then**
        Update n_{zw};
      end
      else
        Update m_{zw};
      end
      Update q_{dz}, p_{zw};
    end
  end
  for $z \leftarrow$ *1 to* T **do**
    Update Ω_z by the method of moments as in Equations 5.6 and 5.7;
  end
end
Compute the posterior estimates of $\alpha, \beta, \gamma, \delta$ defined in Equations 5.2, 5.3, 5.4, 5.5;

Algorithm 2: Inference algorithm for the `NTOT` model

5.2.1 Inference and Parameter Estimation

We adopt collapsed Gibbs sampling in order to do posterior inference. Collapsed Gibbs sampling integrates out irrelevant (nuisance) parameters when conducting inference. This results in a faster inference especially for a complex graphical model as

ours where computational burden at each iteration is reduced considerably compared to the uncollapsed Gibbs sampling technique. In order to estimate the Beta distributions Ω_z we adopt a method of moments where distributions are estimated once per iteration. We present an overview of the collapsed Gibbs sampler in Algorithm 2. We also present the complete posterior inference derivation in Appendix B.

We again describe some notations which will be used later in the text. Let W be the number of words in the vocabulary. $\mathbf{z}$ be the topic variable for the corpus, $z^d_{\neg i}$ is the topic assignment for all words except the current word i. Similar interpretation applies to $x^d_{\neg i}$. Let n_{zw} be the number of times word w has been assigned to z as a unigram; m_{zwv} be the number of times word v has been assigned to z as the second term of a bigram when the previous word is given; p_{zwk} is the number of times the status variable $x = k$ given the previous word and previous word's topic z; q_{dz} is the number of times a word is assigned to topic z in document d. $\bar{t}_z$ is the sample mean. s^2_z is the biased sample variance of the time-stamps which belong to z. $\Omega_{z^d_i 1}$ and $\Omega_{z^d_i 2}$ are shape parameters of the Beta distribution. Count variables also include the assignment of the word being visited. In the collapsed Gibbs sampling procedure, we need to compute the following conditional distribution:

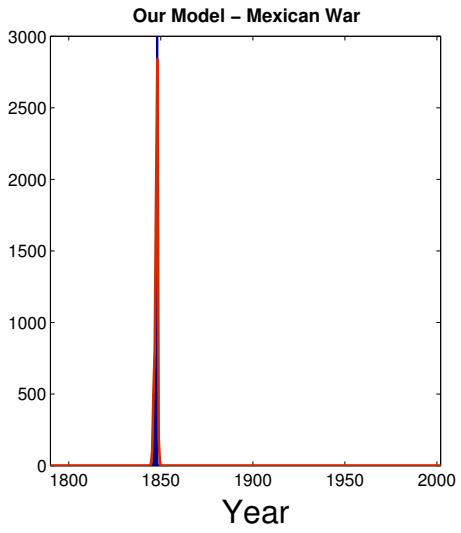
$$P(z^d_i, x^d_i | \mathbf{w}, \mathbf{t}, \mathbf{x}^{\mathbf{d}}_{\neg \mathbf{i}}, \mathbf{z}^{\mathbf{d}}_{\neg \mathbf{i}}, \alpha, \beta, \gamma, \delta, \Omega) \propto \big(\gamma_{x_i{}^d} + p_{z^d_{i-1} w^d_{i-1} x_i} - 1\big)\big(\alpha_{z^d_i} + q_{d z^d_i} - 1\big) \times \underbrace{\frac{(1-t^d_i)^{\Omega_{z^d_i 1}-1} t_i^{d\Omega_{z^d_i 2}-1}}{B(\Omega_{z^d_i 1}, \Omega_{z^d_i 2})}}_{\text{Captures temporal information}} \times \begin{cases} \frac{\beta_{w^d_i} + n_{z^d_i w^d_i} - 1}{\sum_{v=1}^{W}(\beta_v + n_{z^d_i v}) - 1} & \text{if } x^d_i = 0 \\ \frac{\delta_{w^d_i} + m_{z^d_i w^d_{i-1} w^d_i} - 1}{\sum_{v=1}^{W}(\delta_v + m_{z^d_i w^d_{i-1} v}) - 1} & \text{if } x^d_i = 1 \end{cases} \tag{5.1}$$

Simple manipulations help us arrive at the following posterior estimates of $\theta, \phi, \psi, \sigma, \Omega$ which are shown in Equations 5.2,5.3,5.4,5.5,5.6,5.7.

$$\hat{\theta}^d_z = \frac{\alpha_z + q_{dz}}{\sum_{t=1}^{T}(\alpha_t + q_{dt})} \tag{5.2}$$

$$\hat{\phi}_{zw} = \frac{\beta_w + n_{zw}}{\sum_{v=1}^{W}(\beta_v + n_{zv})} \tag{5.3}$$

$$\hat{\psi}_{zwk} = \frac{\gamma_k + p_{zwk}}{\sum_{k=0}^{1}(\gamma_k + p_{zwk})} \tag{5.4}$$



1. east bank	8. military
2. american coins	9. general herrera
3. mexican flag	10. foreign coin
4. separate independent	11. military usurper
5. american commonwealth	12. mexican treasury
6. mexican population	13. invaded texas
7. texan troops	14. veteran troops

Figure 5.3: Histogram along with the high probability n-gram words, shown in a table along side, obtained from our `NTOT` model. The histogram depicts how topics are distributed over time. The histogram is fitted with the Beta PDF. We see that our model generates more localized topics over time with better event-specific n-gram words which makes more sense to a reader than unigram models.

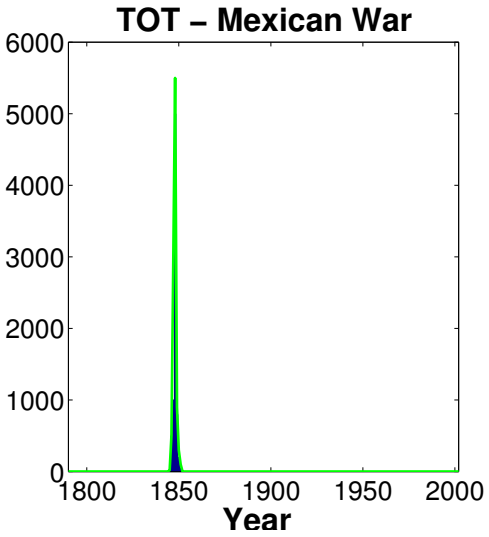
$$\hat{\sigma}_{zwv} = \frac{\delta_v + m_{zwv}}{\sum_{v=1}^{W}(\delta_v + m_{zwv})} \quad (5.5) \qquad \hat{\Omega}_{z1} = \overline{t_z}\left(\frac{\overline{t_z}(1-\overline{t_z})}{s_z^2} - 1\right) \quad (5.6) \qquad \hat{\Omega}_{z2} = (1-\overline{t_z})\left(\frac{\overline{t_z}(1-\overline{t_z})}{s_z^2} - 1\right) \quad (5.7)$$

5.3 Experiments and Results

5.3.1 Data Sets and Comparative Method

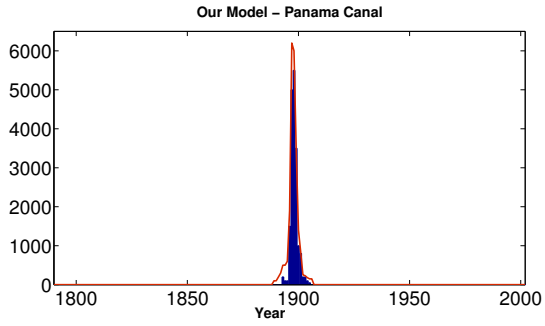
We have conducted experiments on two datasets. Our first dataset comprises of the U.S. Presidential State-of-the-Union[1] speeches from 1790 to 2002. Our second dataset was derived from the NIPS conference papers. The speech dataset and the NIPS paper dataset have also been used in [260]. Some basic information on

[1]http://infomotions.com/etexts/gutenberg/dirs/etext04/suall11.txt



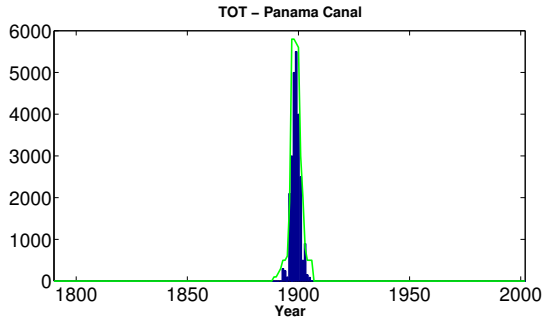
1. mexico	8. territory
2. texas	9. army
3. war	10. peace
4. mexican	11. act
5. united	12. policy
6. country	13. foreign
7. government	14. citizens

Figure 5.4: Histogram along with the high probability n-gram words, shown in a table along side, obtained from our TOT model. The histogram depicts how topics are distributed over time. The histogram is fitted with the Beta PDF. We see that words in the topics are not very insightful when compared with our model.



1. panama canal	8. united states senate
2. isthmian canal	9. french canal company
3. isthmus panama	10. caribbean sea
4. republic panama	11. panama canal bonds
5. united states government	12. panama
6. united states	13. american control
7. state panama	14. canal

Figure 5.5: Histogram along with the high probability n-gram words, shown in a table along side, obtained from our NTOT model. The histogram depicts how topics are distributed over time. The histogram is fitted with the Beta PDF. We see that our model generates more localized topics over time with better event-specific n-gram words which makes more sense to a reader than unigram models.



1. government	8. spanish
2. cuba	9. island
3. islands	10. act
4. international	11. commission
5. powers	12. officers
6. gold	13. spain
7. action	14. rico

Figure 5.6: Histogram along with the high probability n-gram words, shown in a table along side, obtained from our NTOT model. The histogram depicts how topics are distributed over time. The histogram is fitted with the Beta PDF. We see that our model generates more localized topics over time with better event-specific n-gram words which makes more sense to a reader than unigram models.

these datasets can be obtained from [209], [260]. Note that the original raw NIPS dataset[2] consists of 17 years of conference papers. To construct the second dataset, we supplemented this dataset by including some new raw NIPS documents[3] and it has 19 years of papers in total. Our NIPS collection consists of 2740 documents comprising of 45,360,69 non-unique words and 94,961 words in the vocabulary. Our closest comparative method is the TOT model. We have followed the same text pre-processing strategy as in TOT in our both datasets but we maintain the order of terms in documents with stopwords removed. We have fixed the number of topics[4] to 50 and assumed a symmetric Dirichlet distribution ($\alpha = 50/T$ and $\beta = 0.1$) for our model. In addition, in our model we have set $\gamma = 0.01$ and $\delta = 0.01$. For the TOT model, we have fixed the number of topics to 50 and also assumed a symmetric Dirichlet distribution ($\alpha = 50/T$ and $\beta = 0.1$) in all our experiments.

[2]http://www.cs.nyu.edu/~roweis/data.html

[3]http://ai.stanford.edu/~gal/Data/NIPS/

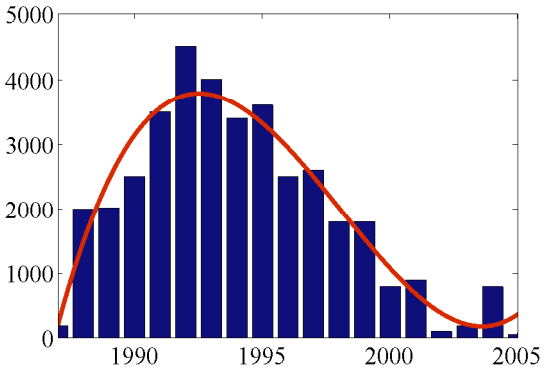
[4]We have used the same number of topics and parameter values as used in the original TOT paper.

5.3.2 Experimental Results

We investigated qualitatively two topics, namely, "Mexican War" and "Panama Canal" from the State-of-the-Union dataset which have also been studied in [260]. Result of our model NTOT for the "Mexican War" is shown in Figure 5.3. In the figure, histogram depicts the way topics are distributed over time, and it is fitted with Beta probability density function. We have shown the top probable words next to the figure. The topic names, for example, "Mexican War" are our own interpretation of the topics. Just like the TOT model in Figure 5.4, from the histogram depiction, our model has also captured the temporal information preciously where we notice that topics are narrowly focused with time based on the timeline when the event occurred. However, the most noticeable observation are the words in each topic. The TOT model captures unigrams where some are ambiguous such as "united" in the "Mexican War" topic. In contrast, our model has produced self-explanatory phrases thereby removing ambiguities. It is interesting to note that unlike TOT, our model has captured some entities popular during that time such as "General Herrera" who was a notable figure during the "Mexican War".

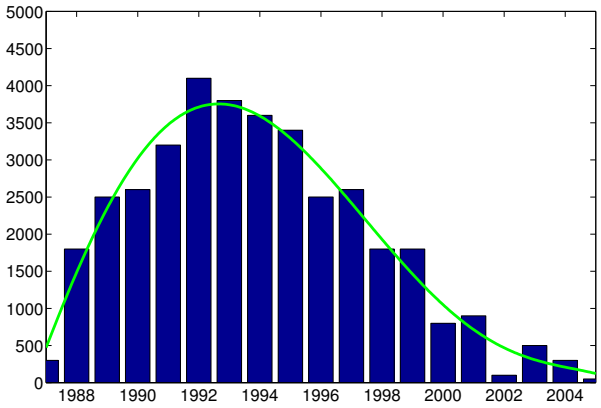
In the topic "Panama Canal" in Figure 5.5, we also capture the same timeline as TOT, shown in Figure 5.6, i.e. from 1904 to 1914 where we note high peaks about this topic during this period. Our results are far more superior with more coherent and interpretable topics. Our model could capture "isthmian canal", "french canal company" etc, which the TOT model could not capture. These entities were popular during that time.

We show a qualitative result of our model using the NIPS collection. We depict the results obtained from our model in Figure 5.7, and the TOT model in Figure 5.8. In order to compute the distribution of topics based on time-stamps, we use Bayes' rule, and compute $E(\theta_{z_i}|t) = P(z_i|t) \propto P(t|z_i)P(z_i)$, where $P(z_i)$ can be assumed to be uniform or estimated from data [260]. We show some of the top probable words



1. hidden unit	6. learning algorithms
2. neural net	7. error signals
3. input layer	8. recurrent connections
4. recurrent network	9. training pattern
5. hidden layers	10. recurrent cascade

Figure 5.7: A topic related to "recurrent NNs" comprising of n-gram words obtained from our model. The title names are given by us based on our interpretation. Histograms depict the way topics are distributed over time, and they are fitted with Beta probability density functions. We have shown the top probable words in topic.



1. state	6. sequences
2. time	7. recurrent
3. sequence	8. models
4. states	9. markov
5. model	10. transition

Figure 5.8: A topic related to "recurrent NNs" comprising of n-gram words obtained from the TOT model. The title names are given by us based on our interpretation. Histograms depict the way topics are distributed over time, and they are fitted with Beta probability density functions. We have shown the top probable words in topic.

from topics conditioned on the time-stamps. From the result, our model has captured words which are more insightful in comparison to the TOT model. For example, in the NIPS-1988; TOT only finds "networks" but our model finds "neural networks". This removes ambiguities which could occur in the former case. In addition, just as in TOT model, our model also begins with "neural networks" and then moves towards "classification" and "regression" topic in the end.

We show one topic from the NIPS collection in Figure 5.9 and compare the result directly with the TOT model whose result is shown in Figure 5.10. Our model has captured localization of topic similar to TOT. However, a major difference lies in the discovered topical phrases with high probability which appear to be more insightful and coherent in our results.

We also perform a quantitative analysis. In [260] the authors showed time-stamp prediction performance of their model in comparison to the LDA model. They had in fact used their alternative TOT model described in the same paper for such prediction. Our model can also be transformed to perform the same prediction task where each time-stamp value i.e. $(t_{i-1}, t_i, t_{i+1}$ etc.) connected with the corresponding latent variables i.e. $(z_{i-1}, z_i, z_{i+1}$ etc.) in the graphical model is removed. Then we assume only a single time-stamp variable t which then can be connected to θ with the arrow head pointing from θ towards t and Ω pointing towards t (i.e. $\theta \rightarrow t$ and $t \leftarrow \Omega$). The time-stamp generation procedure then becomes equivalent to the TOT. However, in contrast to the TOT model, our model computes the time-stamp probabilities of n-gram words from their corresponding topic-wise Beta distributions over time. We show this modified graphical model in Figure 5.11. Unlike [260] we do not discretize the time-stamp as both the models assume a continuous distribution over time-stamps. For simplicity, we again assume the time-stamp probability of the entire n-gram word as the time-stamp of the "head noun". Our goal here is to predict the time-stamp of the document by maximizing the posterior. The posterior is computed by multiplying the time-stamp probability of all phrases from their

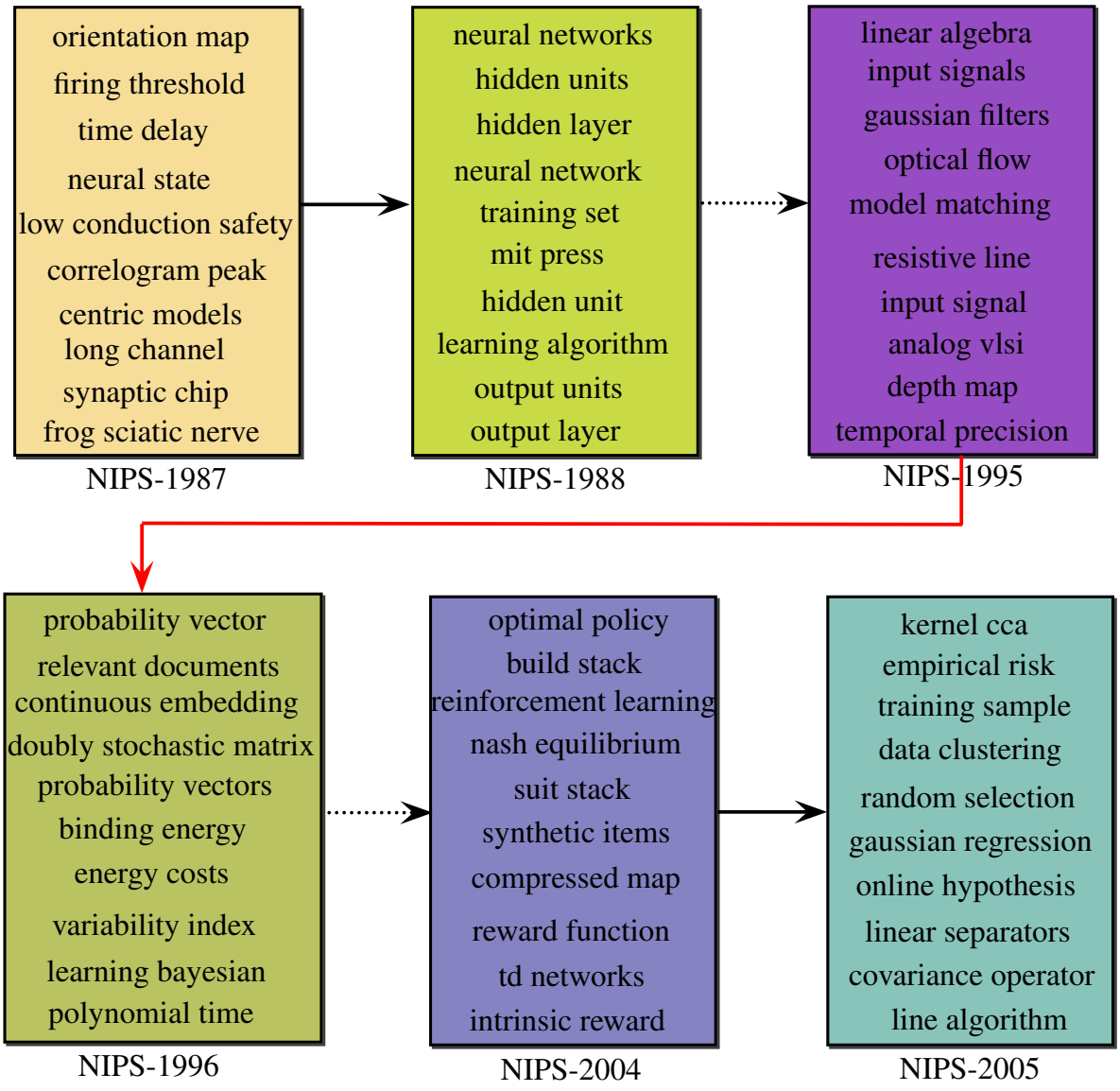


Figure 5.9: An illustration about how topical words change over time as captured by our model. The figure shows top ten probable phrases from the posterior inference in NIPS year-wise. We have only selected some years with some gaps in between, and show top ten phrases/unigrams in that year.

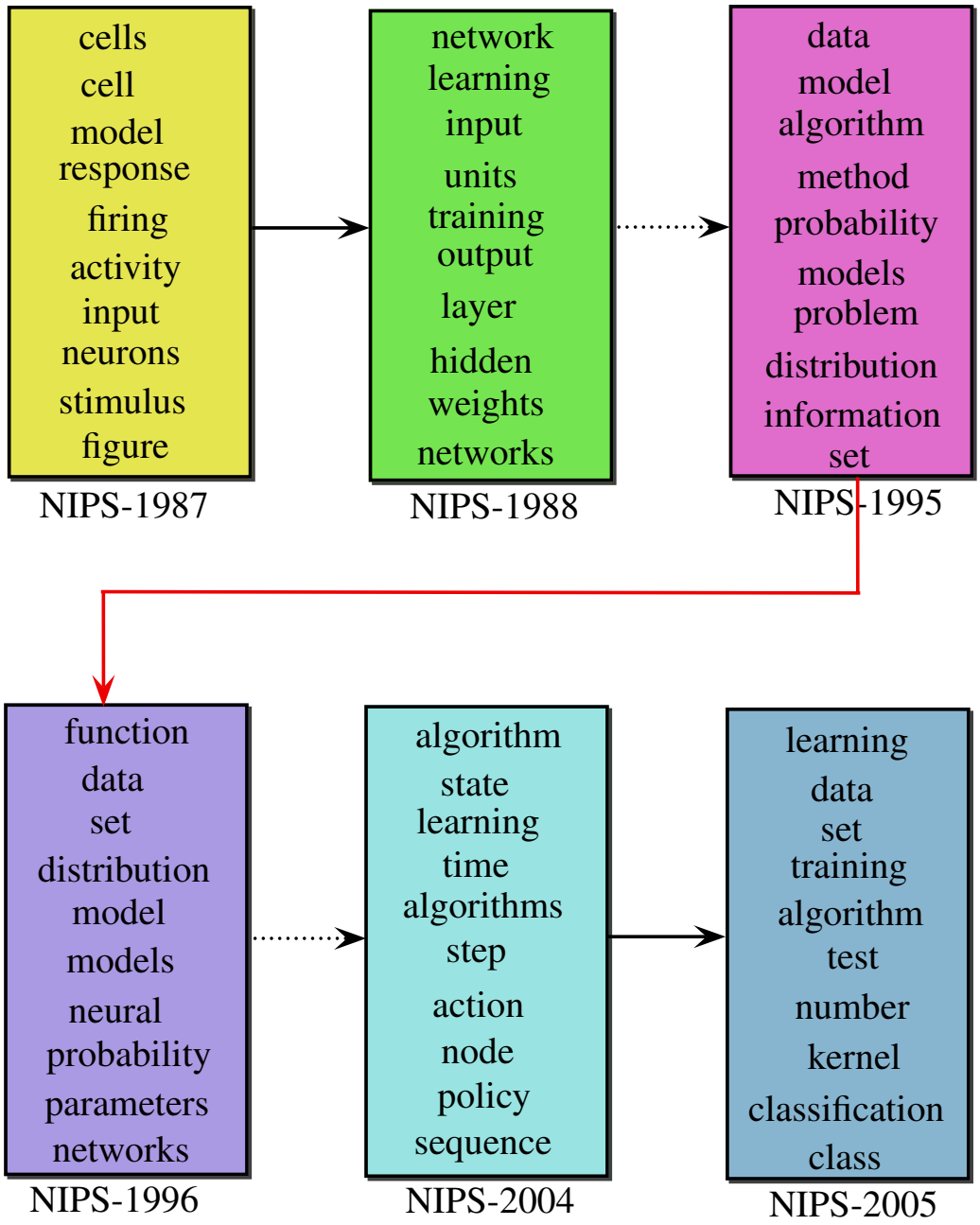


Figure 5.10: An illustration about how topical words change over time as captured by the TOT model. The figure shows top ten probable phrases from the posterior inference in NIPS year-wise. We have only selected some years with some gaps in between, and show top ten unigrams in that year.

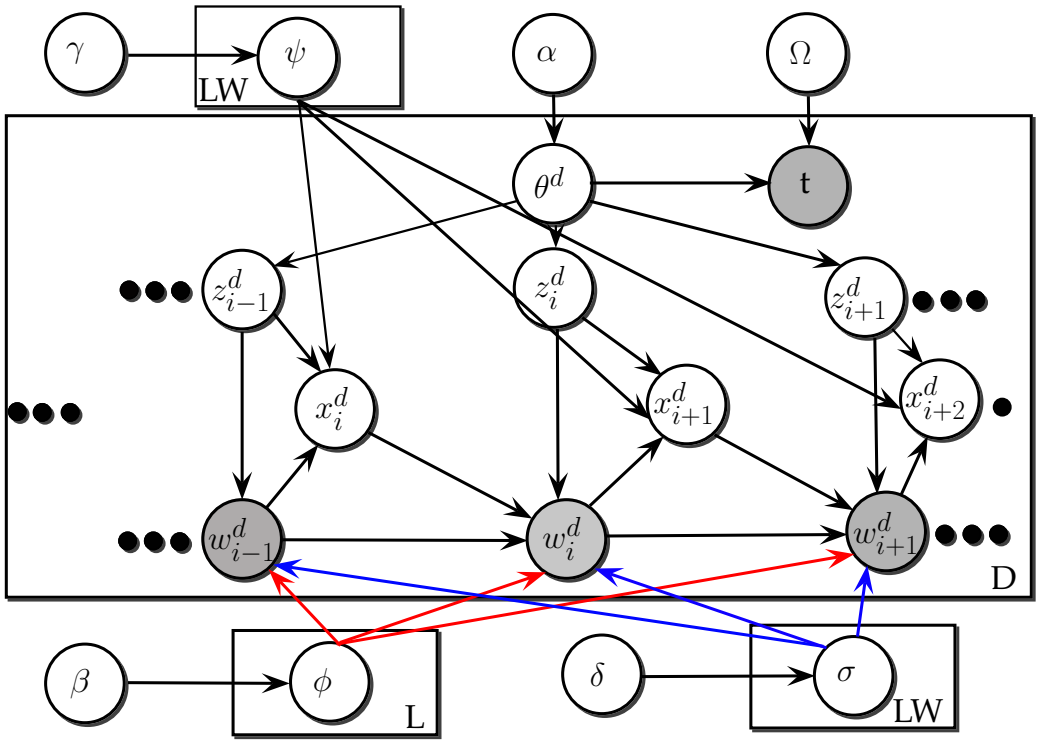


Figure 5.11: A figure showing an alternative view of the NTOT model in standard plate notation. The way time-stamp variable is attached with the document-topic distribution variable makes it appear different from the NTOT model that we have shown previously.

	L1 Error	E(L1)	Accuracy
Our Model	1.60	1.65	0.25
TOT	1.95	1.99	0.20

Table 5.1: Results of decade prediction in the State-of-the-Union speeches dataset.

corresponding topic-wise distributions defined over time. We thus need to compute $\arg\max_t \prod_{i=1}^{N_s^d} P(t|\Omega_{z_i})$ where N_s^d is the number of n-gram words in the document formed by our model. In case of the TOT, we adopt the same posterior computing method as in [260]. We have used the State-of-the-Union dataset and our task is to determine the decade of the new document as adopted in [260]. We have adopted the same three metrics as in [260] and their details are available therein. Comparison results are shown in Table 5.1. Compared to the TOT model, our model achieves better prediction accuracy.

In Figures 5.12 and 5.13, we examine the topic co-occurrences over time for both TOT and NTOT models respectively. As stated in [260], two topics tend to co-occur in the document if the topic proportion for the two topics is greater than certain threshold in that document. We can then count the number of documents for which certain number of topics co-occur. This can help to map how co-occurrence pattern change over time.

From the figures we can see that both the models capture the topical trend almost the same. But our model tends to capture more fine grained topics over time i.e. they tend to spike for a certain time and then diminish which is not the case with the TOT model as it tends to keep on capturing the topics over time despite the popularity of the topic has reduced as time has progressed.

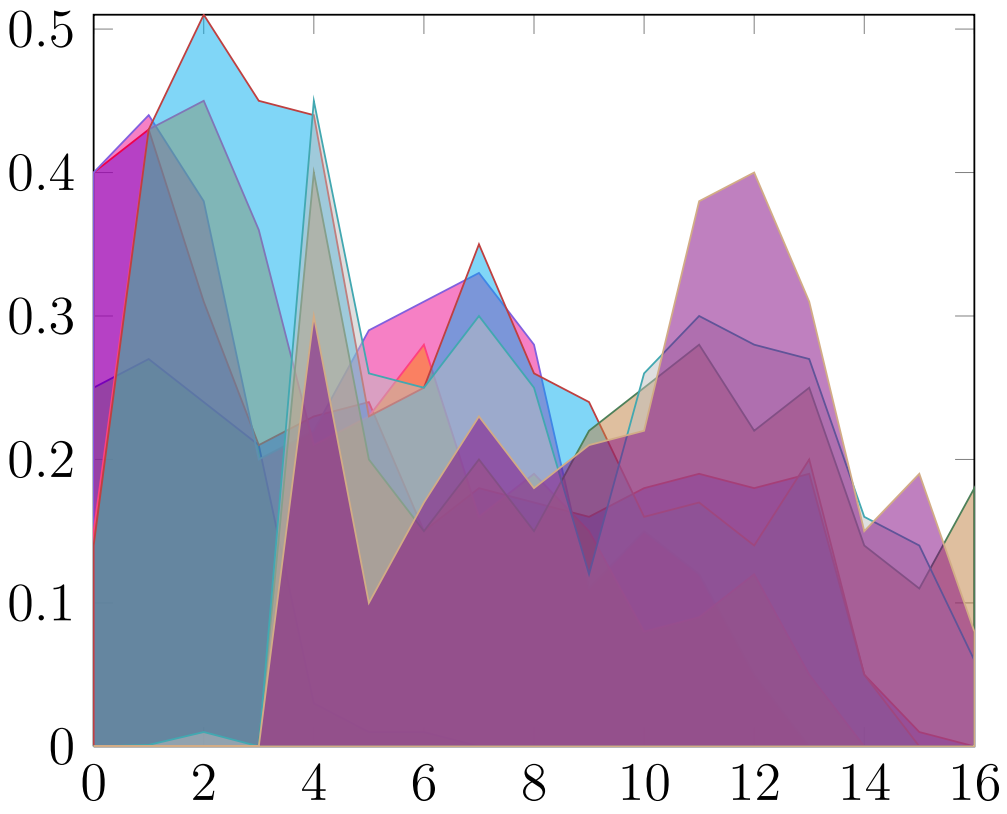


Figure 5.12: A plot showing the co-occurrence of other topics with respect to the "classification" topic in the NIPS dataset for the TOT model. Other topics which co-occur with the "classification" topic can be viewed as lying in the white background of the plot. In the plot, the line with colour — represents Neural Networks topic, line with colour — shows Neural Networks Structure topic, line with colour — shows Distance topic, line with colour — represents the digits topic, line with colour — represents Mixture Models topic, line with colour — represents SVM topic, line with colour — represents Boosting topic, and the line with colour — represents NLP topic.

.

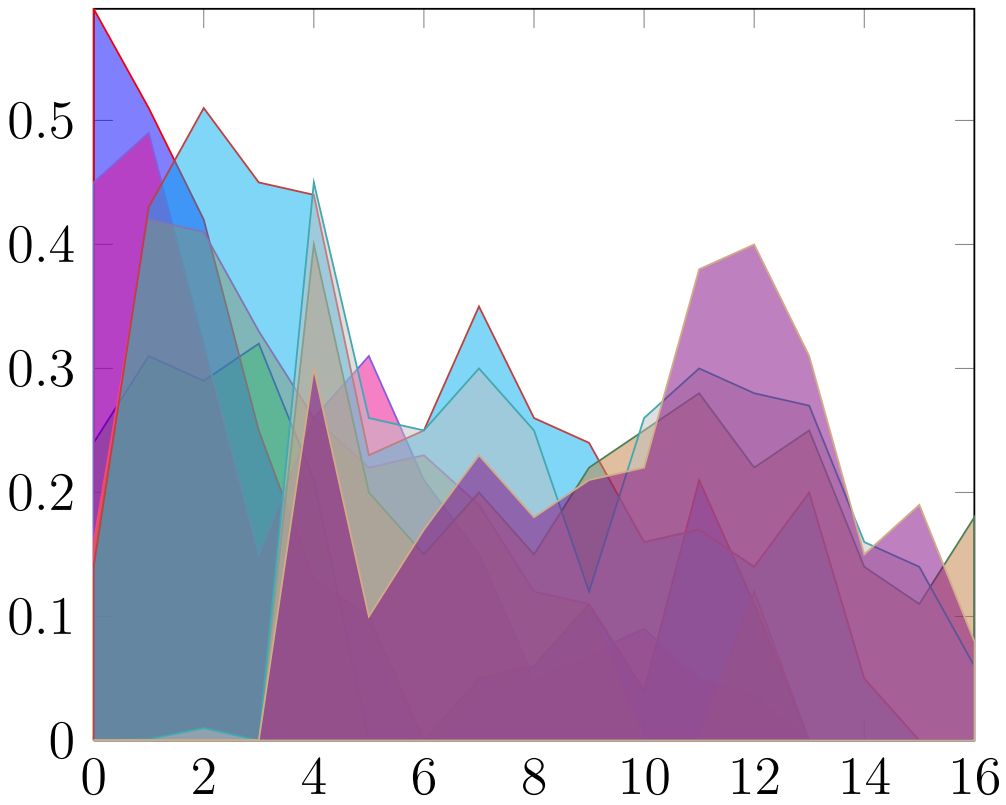


Figure 5.13: A plot showing the co-occurrence of other topics with respect to the "classification" topic in the NIPS dataset for the `NTOT` model. Other topics which co-occur with the "classification" topic can be viewed as lying in the white background of the plot. In the plot, the line with colour — represents Neural Networks topic, line with colour — shows Neural Networks Structure topic, line with colour — shows Distance topic, line with colour — represents the digits topic, line with colour — represents Mixture Models topic, line with colour — represents `SVM` topic, line with colour — represents Boosting topic, and the line with colour — represents NLP topic.

.

5.4 Closing Remarks

We have presented an n-gram topic model which can capture both temporal structure and n-gram words in the time-stamped documents. Topics found by our model are more interpretable with better qualitative and quantitative performance on two publicly available datasets. We have derived a collapsed Gibbs sampler for faster posterior inference. An advantage of our model is that it does away with ambiguities that might appear among the words in topics by considering the temporal dynamics of data.

Chapter Six

Bayesian Nonparametric Topic Models for Text Data

Chapter Summary

In this chapter, we will present three nonparametric topic models with word order that can generate insightful n-gram words in topics. In addition, our models can automatically detect an appropriate number of latent topics from the characteristics of text data. In contrast, most nonparametric topic models such as `HDP`*, when viewed as an infinite-dimensional extension to the* `LDA`*, rely on the bag-of-words assumption. They thus lose the semantic ordering of the words inherent in the text which can give an extra leverage to the computational model, and thus generate less interpretable latent topics.*

6.1 The Case for Bayesian Nonparametric Topic Models With Order

Assuming exchangeability [3] among words (terms) in documents has been the holy-grail in many areas of text processing such as probabilistic topic modeling [23], [233] and many other models which are mainly unigram based topic models. One reason is that such an assumption simplifies the modeling [226] and has an advantage for the computational efficiency [144]. However, this assumption has many disadvantages. One of the main disadvantages is that many unigram words discovered in the latent topics are not very insightful to a reader [166]. Another disadvantage is that the model is not able to take an extra semantic information that is conveyed by the order of the words in the document [117]. This results in an inferior performance in some qualitative and quantitative tasks [136], [8].

Most of the topic models which maintain the order of the terms in the document such as [261], [136], [166], [117], etc. are parametric models. The underlying meaning is that the parameter space is fixed and some parameters, such as the number

of topics, need to be pre-defined by the user. This might be impractical because the user may not always know the true number of latent topics inherent in the data. One way to address this issue is to learn several models with different number of topics and choose the one that has the best performance measure [49]. But this is not a principled approach and it is very time consuming taking up immense computational resources [56]. One way to deal with the problem is to automatically infer a desirable number of latent topics based on the text data characteristics in the document collection. Such models are known as nonparametric probabilistic topic models which are characterized by an infinite-dimensional parameter space. Models such as HDP [247] when used as a topic model[1] can automatically infer the number of latent topics based on the data characteristics, but it assumes exchangeability among words in the documents. It thus inherits some of the limitations of the unigram based topic models.

Considering the above limitations of the existing probabilistic topic models, we propose three new n-gram based topic models for text data that can generate insightful n-gram words in topics. Also, our proposed models can automatically detect an appropriate number of latent topics from the characteristic of text data. Our n-gram nonparametric models assume a First-Order Markovian structure on the order of the words in the documents. By introducing a set of binary random variables in the HDP model and by doing some extra book-keeping during sampling, we can capture topical n-gram words. We also present the corresponding posterior inference schemes for the two models based on the Chinese Restaurant Franchise methods. Our model can also scale to large document collections. We conduct extensive experiments on both small and large publicly available text collections for text mining tasks including document modeling and document classification.

[1]Note that in [247], the authors only introduced the HDP model in general, and not for topic modeling in particular.

6.2 Nonparametric N-gram Collocation Model

We describe our n-gram nonparametric topic model which maintains the order of words, called N-gram Hierarchical Dirichlet Process (NHDP), which is an extension to the basic HDP model described in Chapter 3. Unlike the basic HDP model, our proposed NHDP model is no longer invariant to the reshuffling of words in a document.

We introduce a set of binary random variables $\mathbf{x}$ which we term as the concatenation indicator variable that assume either of the two values which are 0 or 1. This variable indicates whether two words in consecutive order can be concatenated or not. Note that NHDP uses the first order Markov assumption on the words. There are two assignments per word w_i^d at position i in the document d, and $1 \leq i \leq N^d$ where N^d is the number of words (unigrams) in the document d. One assignment is the topic and the other assignment is the concatenation indicator variable x_i^d which relates to whether the word w_i^d can be concatenated with the previous word w_{i-1}^d. If $x_i^d = 1$, then w_i^d is part of a concatenation and the word is generated from a distribution that is dependent only on w_{i-1}^d. x_i^d is drawn from $P(x_i^d|w_{i-1}^d)$. On the other hand, if $x_i^d = 0$, then w_i^d is generated from the distribution associated with its topic. We assume that the first indicator variable x_{d1} in a document is observed and set to 1, and only a unigram is allowed at the beginning of the document. In fact, we can also enforce other constraints in the model. Some examples are: no concatenation is allowed for sentence or paragraph boundary, only a unigram is allowed after a stopword is removed from that position, etc.

Note that NHDP can capture word dependencies in the document. The conditional probability $P(w_i^d|w_{i-1}^d)$ can be written as:

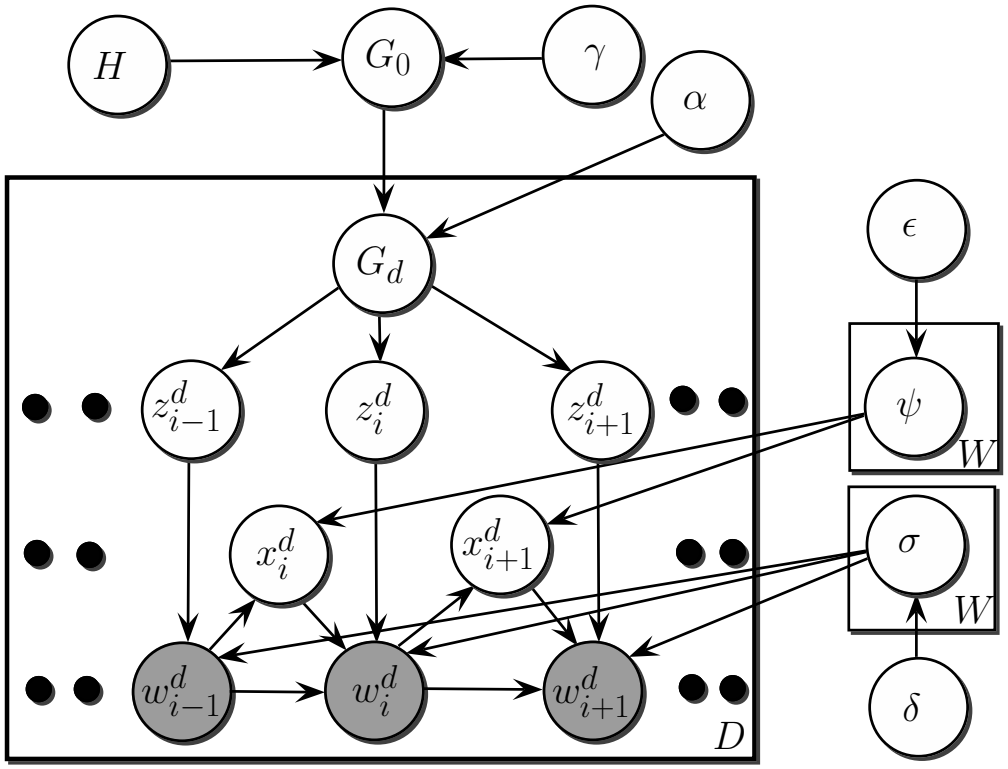


Figure 6.1: Our nonparametric n-gram topic model for generating word collocations shown in a standard plate diagram.

$$P(w_i^d|w_{i-1}^d) = P(w_i^d|w_{i-1}^d, x_i^d = 1)P(x_i^d = 1|w_{i-1}^d) + P(w_i^d|w_{i-1}^d, x_i^d = 0) \\ P(x_i^d = 0|w_{i-1}^d) \qquad (6.1)$$

We can observe that $P(w_i^d|w_{i-1}^d, x_i^d = 0)$ can be computed using the basic HDP model. The full definition of our NHDP model is given as follows:

1. $G_0|\gamma, H \sim \mathbf{DP}(\gamma, H)$
2. $G_d|\alpha, G_0 \sim \mathbf{DP}(\alpha, G_0)$
3. $z_i^d|G_d \sim G_d$
4. $x_i^d|w_{i-1}^d \sim \mathbf{Bernoulli}(\psi_{w_{i-1}^d})$

5. If $x_i^d = 1$ then $w_i^d | w_{i-1}^d \sim$ **Multinomial**$(\sigma_{w_{i-1}^d})$ otherwise $w_i^d | z_i^d \sim F(z_i^d)$

Note that in the definition of our model the hyperprior of σ is δ. The hyperprior value of ψ is ϵ. Just as in the HDP model described earlier, the distribution $F(z_i^d)$, is the Multinomial distribution in the above generative process. We can obtain higher order n-grams by concatenating the current concatenated words with the next n-gram based on the value obtained by the next concatenation indicator variable. Although our model does not directly generate topic-wise n-grams, an n-gram can be associated with a topic via a simple post-processing strategy. One strategy is to take the topic of the first term in the n-gram as the topic for the whole n-gram. This technique has been used in [175] for the LDACOL model. Another strategy is to assume the topic of the n-gram as the most common topic occurring in the words involving in that n-gram [175].

6.3 Posterior Inference

Our inference scheme is based on the Chinese Restaurant Franchise scheme [247] with some modifications. In our scheme, we have to handle two different conditions. The first condition is concerned with $x_i^d = 0$ whereas the second condition is concerned with $x_i^d = 1$. Note that for some observed x_i^d, only z_i^d needs to be drawn.

In the document modeling setting, each document is referred to as a restaurant and words in the document are referred to as customers. The set of documents share a global menu of topics. The words in the document are divided into groups, each of which shares a table. Each table is associated with a topic and words around each table are associated with the table's topic.

6.3.1 The First Condition:

The first condition refers to $x_i^d = 0$. In this setting, most of the modeling will resemble the HDP model as presented in [247], but in our case we need to derive updates for the HDP model for text data.

We will sample t_{di} which is the table index for each word w_i^d at the position i in the document d. We will then sample k_{dt} which is the topic index variable for each table t in d. $k_{d\hat{t}}$ is the new topic index variable created for a new table. Note that we will only sample the index variables here rather than the distributions themselves [56]. We define $\mathbf{w}$ as $(w_i^d : \forall d, i)$ and $\mathbf{w}_{dt}$ as $(w_i^d : \forall i \text{ with } t_{di} = t)$, $\mathbf{t}$ as $(t_{di} : \forall d, i)$ and $\mathbf{k}$ as $(k_{dt} : \forall d, t)$. In addition, we also define $\mathbf{x}$ as $(x_i^d : \forall d, i)$. When a superscript is attached to a set of variables or count, for example, $(\mathbf{k}^{\neg dt}, \mathbf{t}^{\neg di})$, it means that the variables corresponding to the superscripted index are removed from the set or from the calculation of the count. Each word whose $x_i^d = 0$ is assumed to be drawn from $F(z)$ whose density is written as $f(.|\phi)$ (f is just one part obtained from F). This density is the multinomial distribution with the parameter ϕ. The likelihood of w_i^d for $t_{di} = t$ where t is an existing table, denoted as $f_k^{\neg w_i^d}(w_i^d)$, is the conditional density of w_i^d given all words in topic k except w_i^d:

$$f_k^{\neg w_i^d}(w_i^d) = \frac{\int f(w_i^d|\phi_k) \prod_{d'i' \neq di, z_{d'i'}=k} f(w_{d'i'}|\phi_k) h(\phi_k) \mathrm{d}\phi_k}{\int \prod_{d'i' \neq di, z_{d'i'}=k} f(w_{d'i'}|\phi_k) h(\phi_k) \mathrm{d}\phi_k} \tag{6.2}$$

where h is a probability density function of H and H is a Dirichlet distribution over a fixed vocabulary of size W. $h(.)$ is the Dirichlet distribution with the parameter η. ϕ_k is one of the global topics with which each table is associated which is indicated with a table-specific topic index k_{dt}. Furthermore, Equation 6.9 can be simplified as:

$$f_k^{\neg w_i^d}(w_i^d = \vartheta) = \frac{n_{..k}^{\neg w_i^d, \vartheta} + \eta}{n_{..k}^{\neg w_i^d} + W\eta} \tag{6.3}$$

where $n_{..k}^{\neg w_i^d}$ is the number of words belonging to the topic k in the corpus whose $x_i^d = 0$ excluding w_i^d. $n_{..k}^{\neg w_i^d, \vartheta}$ is the number of times the word ϑ is assigned with the

topic k excluding w_i^d and whose x_i^d is 0. Furthermore, W is the number of words in the vocabulary which is typically fixed and is known. The likelihood of w_i^d for $t_{di} = \hat{t}$, where $\hat{t}$ is the new table being sampled, is written as:

$$P(w_i^d|t_{di} = \hat{t}, \mathbf{t}^{\neg di}, \mathbf{k}) = \sum_{k=1}^{L} \frac{m_{.k}}{m_{..} + \gamma} f_k^{\neg w_i^d}(w_i^d) + \frac{\gamma}{m_{..} + \gamma} f_{\hat{k}}^{\neg w_i^d}(w_i^d) \tag{6.4}$$

where $\hat{k}$ is the new topic being sampled. $m_{.k}$ is the number of tables belonging to the topic k in the corpus. $m_{..}$ is the total number of tables in the corpus. $f_{\hat{k}}^{\neg w_i^d}(w_i^d) = \int f(w_i^d|\phi)h(\phi)\mathrm{d}\phi$ is the prior density of w_i^d. γ is the concentration parameter as described. Since we follow the standard Chinese Restaurant Franchise sampling procedure, the conditional density for t_{di} for Gibbs sampling, the conditional densities for $k_{d\hat{t}}$ and k_{dt} can be found in [56].

6.3.2 The Second Condition:

The second condition refers to $x_i^d = 1$. We only need to sample the probability of a topic in a document as the current word w_i^d is generated by the previous word w_{i-1}^d. In order to do this, we proceed as follows:

$$P(k_{dt} = k|\mathbf{t}, \mathbf{k}^{\neg dt}) \propto \begin{cases} m_{.k}^{\neg dt} f_k^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt}) & \text{if } k \text{ is already used} \\ \gamma f_{\hat{k}}^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt}) & \text{if } k = \hat{k} \end{cases} \tag{6.5}$$

where $f_k^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt})$, which is the conditional density of $\mathbf{w}_{dt}$ given all words associated

with the topic k leaving out $\mathbf{w}_{dt}$ is defined as:

$$f_k^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt}) = \frac{\Gamma(n_{..k}^{\neg \mathbf{w}_{dt}} + W\eta)}{\Gamma(n_{..k}^{\neg \mathbf{w}_{dt}} + n^{\mathbf{w}_{dt}} + W\eta)} \times \frac{\prod_{\vartheta} \Gamma(n_{..k}^{\neg \mathbf{w}_{dt},\vartheta} + n^{\mathbf{w}_{dt},\vartheta} + \eta)}{\prod_{\vartheta} \Gamma(n_{..k}^{\neg \mathbf{w}_{dt},\vartheta} + \eta)} \tag{6.6}$$

where $n^{\mathbf{w}_{dt}}$ is the total number of words at the table t whose $x_i^d = 0$. $n^{\mathbf{w}_{dt},\vartheta}$ is the number of times the word ϑ appears at the table t with the assignment $x_i^d = 0$. $n_{..k}^{\neg \mathbf{w}_{dt}}$ is the number of words belonging to topic k in the corpus except $\mathbf{w}_{dt}$.

6.3.3 Sampling the Concatenation Indicator Variables:

We present how to sample the values of the indicator variables. The idea is to compute the probabilities of how often two words consecutively occur in sequence. Then based on the probability value, the indicator variable is set to either 0 or 1. Let $n_0^{w_{i-1}^d}$ and $n_1^{w_{i-1}^d}$ be the number of times word w_{i-1}^d has been drawn from a topic or formed a part of a concatenation respectively and all counts exclude the current case. ϵ_0 and ϵ_1 are the priors of the binomial distribution. $n_{w_i^d}^{w_{i-1}^d}$ is the number of times the word w_i^d comes after the word w_{i-1}^d. $n_{..k}^{\neg w_i^d,\vartheta}$ and $n_{..k}^{\neg w_i^d}$ have been defined in Equation 6.3.

$$P(x_i^d = 0|\mathbf{x}_{\neg di}, \mathbf{w}, \mathbf{k}) \propto \frac{n_0^{w_{i-1}^d} + \epsilon_0}{\sum_{c=0}^{1} n_c^{w_{i-1}^d} + \epsilon_0 + \epsilon_1} \times \frac{n_{..k}^{\neg w_i^d,\vartheta} + \eta}{n_{..k}^{\neg w_i^d} + W\eta} \tag{6.7}$$

$$P(x_i^d = 1|\mathbf{x}_{\neg di}, \mathbf{w}, \mathbf{k}) \propto \frac{n_1^{w_{i-1}^d} + \epsilon_1}{\sum_{c=0}^{1} n_c^{w_{i-1}^d} + \epsilon_0 + \epsilon_1} \times \frac{n_{w_i^d}^{w_{i-1}^d} + \delta}{\sum_{v=1}^{W} n_v^{w_{i-1}^d} + W\delta} \tag{6.8}$$

where δ is same as described in Section 6.2.

6.4 Nonparametric N-gram Topic Models (NNTM)

In this section, we present a detailed description of our two proposed nonparametric topic models called Nonparametric N-gram Topic Model (NNTM). In our model design, we try to address the following key questions:

1. How can we capture n-gram words without breaking the exchangeability assumption so that we can take advantage of the sampling schemes of unigram based HDP model?

2. Although it has been consistently stated in the literature that n-gram based models are very computationally expensive and not easy to apply on large scale datasets, one investigation in this paper is how to make our scheme applicable to large datasets in a nonparametric setting?

HPYP priors are being widely used to capture longer order n-grams such as [134], [166], but these models are impractical for large datasets [9]. Thus new techniques have to be investigated for large scale text data.

In our two proposed nonparametric n-gram topic models with word order, we maintain most of the properties of the unigram based nonparametric topic model, and introduce some extra book-keeping for capturing n-gram words. As we shall show later in our empirical analysis, such extra book-keeping effort does not significantly impact the complexity of our proposed models. We will present our new Chinese Restaurant Franchise scheme with Buddy (a friend) Customers where two friends always take the same table in the restaurant. We first describe our first proposed nonparametric n-gram topic model which we name as NNTM-1. We also present some advantages and disadvantages of our first proposed model. Then we extend the proposed model and propose NNTM-2.

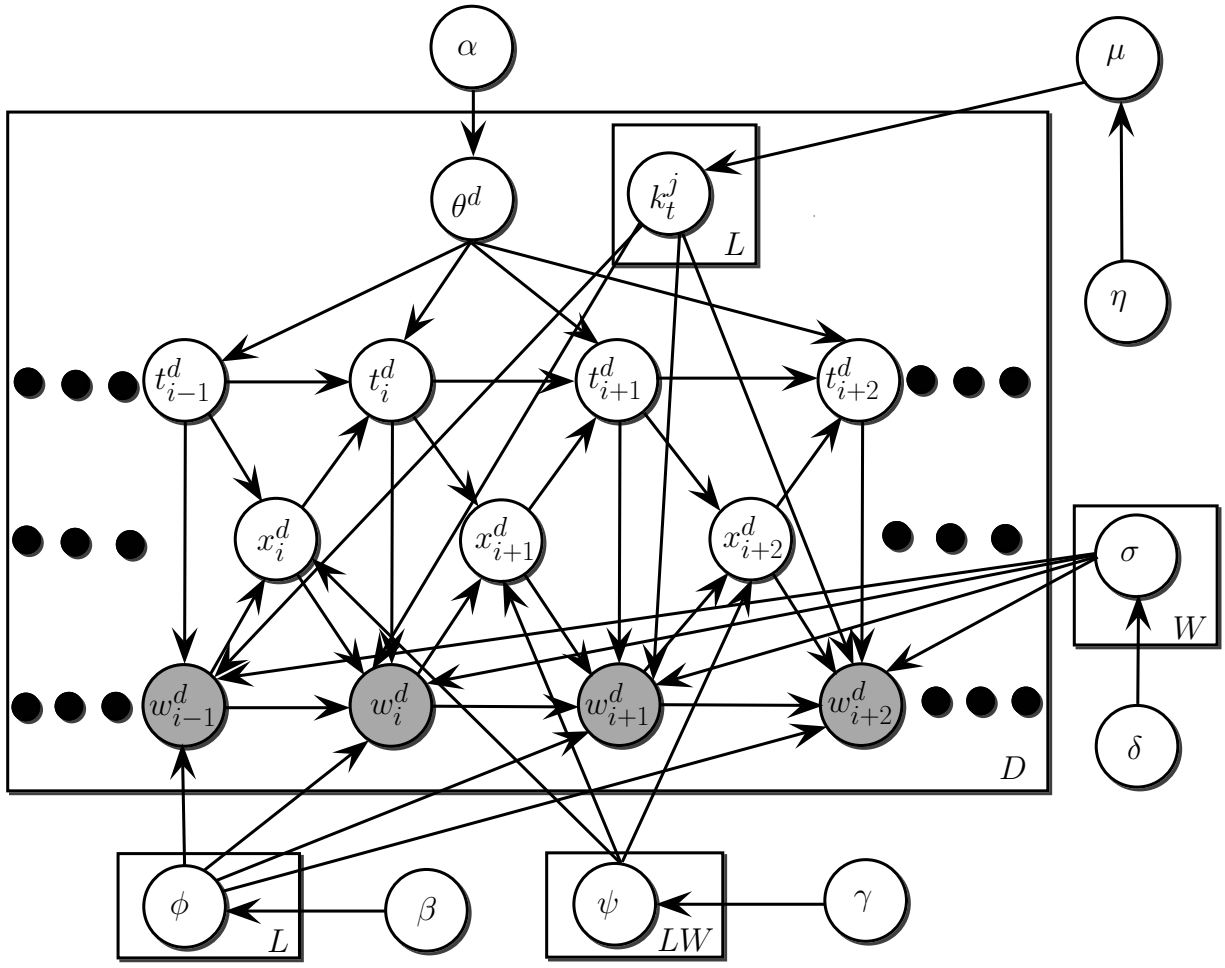


Figure 6.2: Our `NNTM-1` model in the `CRF` scheme with Buddy Customers.

6.4.1 Model Description of NNTM-1

Our another proposed Nonparametric N-gram Topic Model, which we name as `NNTM-1` is a nonparametric topic model for n-gram generation in topics. We show the graphical model of our proposed model in Figure 6.2. The key idea of our model is to keep track of the word order in the document. In order to capture n-grams in topics, we introduce a binary random variable $\mathbf{x}$ in the model. We name this random variable as the concatenation indicator variable that assumes either of the two values which are 0 or 1. This variable indicates whether two words in consecutive order can be concatenated or not to form a bigram. The model uses the first order Markov

assumption on the words. There are two assignments per word w_i^d at the position i in the document d, and $1 \leq i \leq N_d$ where N_d is the number of words (unigrams) in the document d. One assignment is the topic and the other assignment is the concatenation indicator variable which relates to whether the word w_i^d can be concatenated with the previous word w_{i-1}^d. If $x_i^d = 1$, then w_i^d is part of a concatenation, and the word is generated from a distribution that is dependent on w_{i-1}^d. On the other hand, if $x_i^d = 0$, then w_i^d is generated from the distribution associated with its topic, just like the HDP model. x_i^d is drawn from $P(x_i^d | w_{i-1}^d, z_{i-1}^d)$. We assume that the first indicator variable x_1^d in a document is observed and set to 1 (this is analogues to the first customer enters the restaurant and chooses the first table), and only a unigram is allowed at the beginning of the document. In fact, we can also enforce other constraints in the model. One constraint is that no concatenation is allowed for sentence or paragraph boundary. Another constraint is that only a unigram is allowed after a stopword is removed from that position, etc.

The key idea in our model is to store the word order information separately as a three tuple comprising of $((w_i^d, w_{i-1}^d), d, (0, 1))$, which can be represented as a sparse matrix where we can know whether the word w_i^d is preceded by the word w_{i-1}^d in the document d. We set the value 1 when w_i^d precedes w_{i-1}^d otherwise it is 0 and is removed from the sparse representation to conserve storage space. In this way, we do not need a strict constraint that customers enter the restaurant in an order of the occurrence of words in the document. This position matrix, following the first order Markovian assumption, only stores the bigram information. So during the buddy assignment, we only need to know which word preceded the other.

Generative Process

We show the generative process of our first nonparametric n-gram model, `NNTM-1` in the CRF scheme below:

1. Draw ϕ from $H(\beta)$
2. Draw μ from **GEM**(η)
3. Draw **Discrete**(σ) from **Dirichlet**(δ)
4. Draw **Bernoulli**(ψ) from **Beta**(γ)
5. For each document d
   (a) Draw $\widetilde{\theta}^d$ from α
   (b) Draw k_t^d from μ
   (c) For each word w_i^d at position i in the document d
      i. Draw t_i^d from $\widetilde{\theta}^d$ if $x_i^d = 0$ otherwise $t_i^d = t_{i-1}^d$
      ii. Draw x_i^d from **Bernoulli**$(\psi_{t_{i-1}^d w_{i-1}^d})$
      iii. Draw w_i^d from $\phi_{k_{t_i^d}^d}$ if $x_i^d = 0$ else draw $\sigma_{w_{i-1}^d}$

Posterior Inference

To find the latent variables that best explain the observed data, we use Gibbs sampling, a widely used Markov Chain Monte Carlo inference technique [190]. However, inference for nonparametric collocation model is more complicated than it is for parametric models. Teh et al., [247] described three distinct approaches to Gibbs sampling for the HDP including the one based on the CRF scheme. We adopt this scheme in our model with some modifications. In our scheme, we have to handle two different conditions. The first condition is concerned with $x_i^d = 0$ whereas the second condition is concerned with $x_i^d = 1$.

In the document modeling setting, each document is referred to as a restaurant and words in the document are referred to as customers. The set of documents share

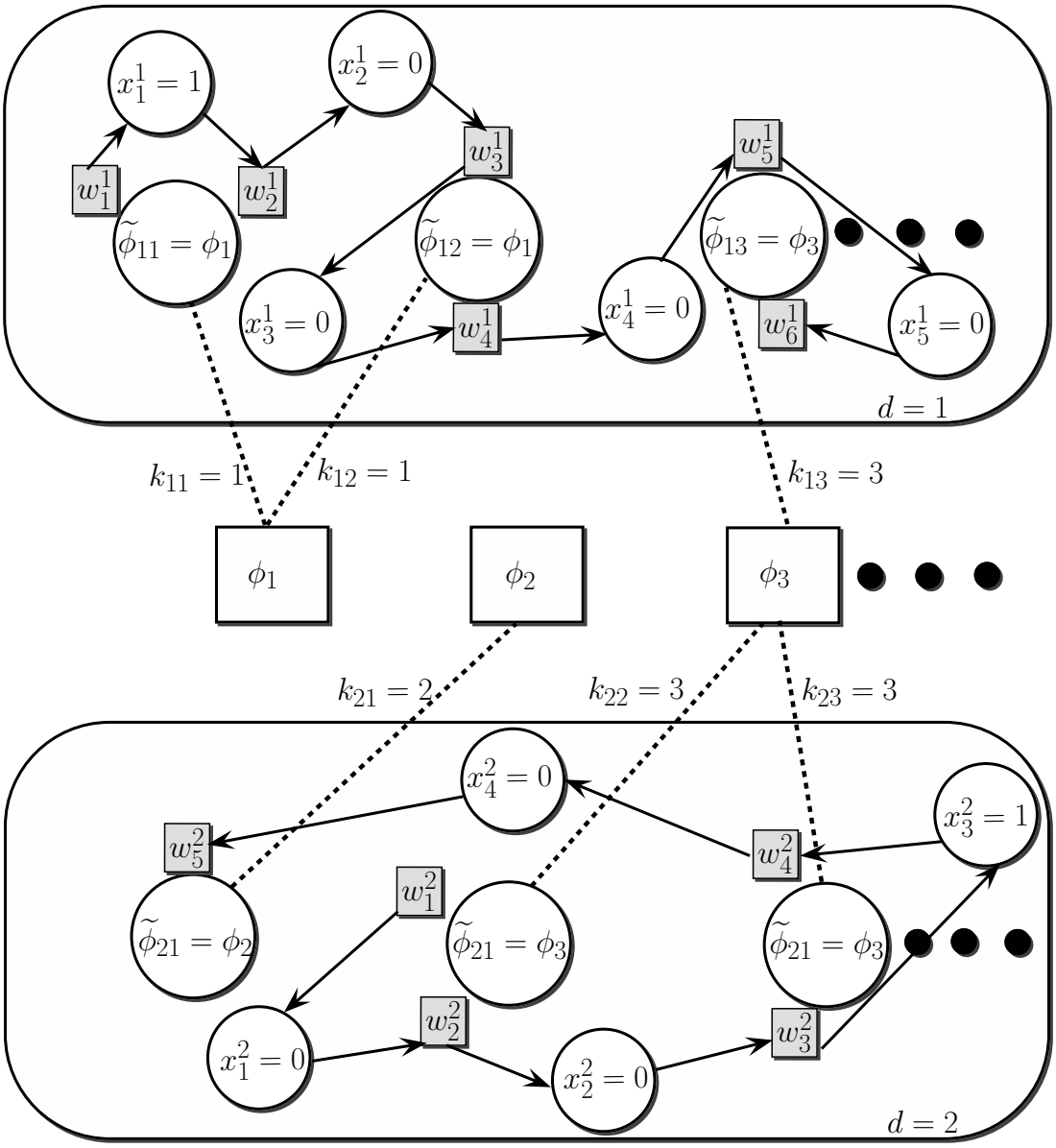


Figure 6.3: An illustration of the CRF scheme incorporating Buddy Customers. The figure depicts the state when all the customers are seated together either alone or with their buddies in the tables. This illustration is an extension of the CRF scheme presented in for the HDP model [239]. As one can see that the variable responsible for buddy allocation is $\mathbf{x}$.

a global menu of topics. The words in the document are divided into groups, each of which shares a table. Each table is associated with a topic and words around each table are associated with the table's topic. $\mathbf{CRP}(\alpha)$ is the distribution based on the Chinese Restaurant Process prior with parameter α. The two-step generative process which will be used for Gibbs sampling for `NNTM-1` model can be written as:

1. {Partitioning} For each document $d \in D$
   (a) For each word at position $i \in d$
      i. Draw the table index $t_i^d | t_1^d, t_2^d, \cdots, t_{i-1}^d, \alpha \sim \mathbf{CRP}(\alpha)$ with $x_i^d = 0$ otherwise set the table index $t_i^d = t_{i-1}^d$
   (b) For each table $t \in d$
      i. Draw the topic $k_t^d | k_1^1, k_2^1, \cdots, k_1^2, k_{t-1}^d, \gamma \sim \mathbf{CRP}(\gamma)$
2. {Generating} For each document $d \in D$
   (a) For each word at position $i \in d$
      i. Draw the bigram indicator variable $x_i^d \sim \psi_{t_{i-1}^d w_{i-1}^d}$
      ii. Draw a word $w_i^d \sim \phi_{k_{t_i^d}^d}$ when $x_i^d = 0$ else draw a word $w_i^d \sim \sigma_{w_{i-1}^d}$

One can note from the algorithm above that our model is a hybrid between the bigram language model and the HDP in the CRF scheme. For instance, when all x's are 0 i.e. $\mathbf{x} = 0$ in a document, then there is no ordering the terms and the process simply follows the HDP sampling procedure. However, when all $\mathbf{x}$'s in the document are 1 i.e. $\mathbf{x} = 1$ in a document, then words are generated only by the previous word. Generation of n-grams in our model is realized by the set of $\mathbf{x}$ in the model. Therefore, if we provide accurate information to this binary random

variable about n-gram generation, it can effectively generate n-grams in addition to topical words generation. Therefore, if we can keep such word order information in an external matrix which can be consulted during the Gibbs sampling procedure, it will help cluster n-grams at the tables along with other words, and still the property of exchangeability remains. Therefore, n-grams with the same topic will always sit together and thus we term this scheme as Buddy Chinese Restaurant Franchise.

The metaphor of the Chinese Restaurant Franchise with Buddy Customers (CRF-BC) can be defined as follows. Consider a restaurant with an infinite set of tables. These tables are analogues to clusters. There is a restaurant franchise with a shared menu which is shared across the restaurants. In each restaurant, at each table, one dish is ordered from the menu by the first customer who takes that table. That dish will now be shared among all other customers who sit at the table. In this scheme, customers are analogues to observations. Multiple tables in different restaurants can serve the same dish. When the next customer arrives at the restaurant, the customer tries to find his/her buddy who might be already sitting there. If the customer finds his/her buddy, the customer sits at the same table as that of the buddy otherwise either sits at any of the other table occupied by the other customers, or chooses a new table altogether. So the customer, who now finds his/her buddy, vacates the table where he/she was sitting and chooses another table along with the buddy. Instances may also arise when the buddy customers upon the arrival of the buddy, may decide to take a new table together. In the end, all buddies sit together in the same table. We show this metaphor diagrammatically in Figure 6.3.

Another way to effectively capture word order in a nonparametric setting would be to use a different metaphor known as
Distance Dependent Chinese Restaurant Franchise (ddCRF) [135], which is an extension to the
Distance Dependent Chinese Restaurant Process (ddCRP) [17]. In this scheme, instead of customers sitting at the tables with other customers, customers are assigned

to other customers or not assigned to anyone. The probability of a new customer being assigned to other customers already sitting is proportional to a decreasing function of the distance between the customers already sitting with the new customer. The connected customers implicitly exhibit a clustering property. Our metaphor can effectively handle unigram and bigram words and assign them to appropriate tables whereas in the ddCRF scheme, n-gram will tend to sit at one table and unigrams belonging to the same cluster will tend to sit at the other tables. This is primarily because connected components (two words in sequence where $x = 1$) sit together as this co-occurrence will determine the seating arrangement. Nevertheless, one can use the second (customer) level CRP as ddCRP in our model to generate topical n-grams. Inquisitive readers are requested to consult [135] for more details. In fact, in our CRF-BC scheme, we also make modifications to the second level CRP in order to handle buddy assignments.

The First Condition

The first condition refers to $x_i^d = 0$. We will sample t_i^d which is the table index for each word w at the position i in the document d. We will then sample k_t^d which is the topic index variable for each table t in d. Note that we will only sample the index variables here rather than the distributions themselves [56]. We define $\mathbf{w}$ as $(w_i^d : \forall d, i)$ and $\mathbf{w}_t^d$ as $(w_i^d : \forall i \text{ with } t_i^d = t)$, $\mathbf{t}$ as $(t_i^d : \forall d, i)$ and $\mathbf{k}$ as $(k_t^d : \forall d, t)$. When a sign $\neg$ in the superscript is attached to a set of variables or count, for example, $(\mathbf{k}^{\neg dt}, \mathbf{t}^{\neg di})$, it means that the variables corresponding to the superscripted index is removed from the set or from the calculation of the count. The likelihood of w_i^d for $t_i^d = t$ where t is an existing table, denoted as $f_k^{\neg w_i^d}(w_i^d)$, is the conditional density of w_i^d given all words in the topic k except w_i^d:

$$f_k^{\neg w_i^d}(w_i^d) = \frac{\int f(w_i^d|\phi_k) \prod_{d'i' \neq di, z_{d'i'}=k} f(w_{d'i'}|\phi_k) h(\phi_k) \mathrm{d}\phi_k}{\int \prod_{d'i' \neq di, z_{d'i'}=k} f(w_{d'i'}|\phi_k) h(\phi_k) \mathrm{d}\phi_k} \tag{6.9}$$

where each word whose $x_i^d = 0$ is assumed to be drawn from $F(\widetilde{\theta}^d)$ whose density is written as $f(.|\phi)$. This density is the multinomial distribution with the parameter ϕ and $h(.)$ is the Dirichlet distribution with the parameter η. Furthermore, Equation 6.9 can be simplified as:

$$f_k^{\neg w_i^d}(w_i^d = \vartheta) = \frac{n_{..k}^{\neg w_i^d, \vartheta} + \eta}{n_{..k}^{\neg w_i^d} + W\eta} \tag{6.10}$$

where $n_{..k}^{\neg w_i^d}$ is the number of words belonging to the topic k in the corpus whose $x_i^d = 0$ excluding w_i^d. $n_{..k}^{\neg w_i^d, \vartheta}$ is the number of times the word ϑ is assigned with the topic k excluding w_i^d and whose x_i^d is 0. Furthermore, W is the number of words in the vocabulary which is typically fixed and is known. The likelihood of w_i^d for $t_i^d = \hat{t}$, where $\hat{t}$ is the new table being sampled, is written as:

$$P(w_i^d|t_i^d = \hat{t}, \mathbf{t}^{\neg di}, \mathbf{k}) = \sum_{k=1}^{L} \frac{m_{.k}}{m_{..} + \gamma} f_k^{\neg w_i^d}(w_i^d) + \frac{\gamma}{m_{..} + \gamma} f_{\hat{k}}^{\neg w_i^d}(w_i^d) \tag{6.11}$$

where $\hat{k}$ is the new topic being sampled. $m_{.k}$ is the number of tables belonging to the topic k in the corpus. $m_{..}$ is the total number of tables in the corpus. $f_{\hat{k}}^{\neg w_i^d}(w_i^d) = \int f(w_i^d|\phi) h(\phi) \mathrm{d}\phi$ is the prior density of w_i^d. The conditional density for t_i^d is:

$$P(t_i^d = t|\mathbf{t}^{\neg di}, \mathbf{k}) \propto \begin{cases} n_{dt.}^{\neg w_i^d} f_{k_{dt}}^{\neg w_i^d}(w_i^d) & \text{if } t \text{ is already used} \\ \alpha P(w_i^d|t_i^d = \hat{t}, \mathbf{t}^{\neg di}, \mathbf{k}) & \text{if } t = \hat{t} \end{cases} \tag{6.12}$$

where $n_{dt.}^{\neg w_i^d}$ is the number of words in the document d at the table t whose $x_i^d = 0$ excluding the current word.

If $t_i^d = \hat{t}$, then $k_{d\hat{t}}$ has to be sampled:

$$P(k_{d\hat{t}} = k|\mathbf{t}, \mathbf{k}^{\neg d\hat{t}}) \propto \begin{cases} m_{.k} f_k^{\neg w_i^d}(w_i^d) \text{ if } k \text{ is already used} \\ \gamma f_{\hat{k}}^{\neg w_i^d}(w_i^d) & \text{if } k = \hat{k} \end{cases} \tag{6.13}$$

As described in [258], to sample k_{dt} for each table in each document, we compute the conditional density of $\mathbf{w}_{dt}$ ($\mathbf{w}_{dt}$ is defined as all the words at the table t in the document d) given all words assigned to the topic k excluding $\mathbf{w}_{dt}$:

$$P(k_{dt} = k|\mathbf{t}, \mathbf{k}^{\neg dt}) \propto \begin{cases} m_{.k}^{\neg dt} f_k^{\mathbf{w}_{dt}}(\mathbf{w}_{dt}) & \text{if } k \text{ is already used} \\ \gamma f_{\hat{k}}^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt}) & \text{if } k = \hat{k} \end{cases} \tag{6.14}$$

where

$$f_k^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt}) = \frac{\Gamma(n_{..k}^{\neg \mathbf{w}_{dt}} + W\eta)}{\Gamma(n_{..k}^{\neg \mathbf{w}_{dt}} + n^{\mathbf{w}_{dt}} + W\eta)} \times \frac{\prod_{\vartheta} \Gamma(n_{..k}^{\neg \mathbf{w}_{dt},\vartheta} + n^{\mathbf{w}_{dt},\vartheta} + \eta)}{\prod_{\vartheta} \Gamma(n_{..k}^{\neg \mathbf{w}_{dt},\vartheta} + \eta)} \tag{6.15}$$

where $n^{\mathbf{w}_{dt}}$ is the total number of words at the table t whose $x_i^d = 0$. $n^{\mathbf{w}_{dt},\vartheta}$ is the number of times the word ϑ appears at the table t with the assignment $x_i^d = 0$.

The Second Condition

In this condition, the second term of a bigram shares the same topic as the first term. It means that the second term simply enters the restaurant and sits at the table where the buddy customer is already sitting. This can be expressed as:

$$P(t_i^d = t|\mathbf{t}^{\neg di}, \mathbf{k}) \propto (t_i^d = t_{i-1}^d) \tag{6.16}$$

Sampling the Concatenation Indicator Variables

We present how to sample the values of the concatenation indicator variables. These variables will determine the buddy assignment. The idea is to compute the probabilities of how often two words consecutively occur in sequence. Then based on the probability value, the indicator variable is set to either 0 or 1. Let $p_{t_{i-1}^d w_{i-1}^d x}$ be the number of times the concatenation indicator variable x_i^d has been set to 0 or 1 given the previous word and the topic of the previous word. Note that a table is associated with a topic. $n_{w_i^d}^{w_{i-1}^d}$ is the number of times the word w_i^d comes after the

word w_{i-1}^d in the entire corpus.

$$P(x_i^d = 0|\mathbf{x}^{\neg di}, \mathbf{w}, \mathbf{t}) = \frac{p_{t_{i-1}^d w_{i-1}^d 0} + \gamma_0}{\sum_{c=0}^{1} p_{t_{i-1}^d w_{i-1}^d c} + \gamma_0 + \gamma_1} \times f_k^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt}) \tag{6.17}$$

$$P(x_i^d = 1|\mathbf{x}^{\neg di}, \mathbf{w}, \mathbf{t}) = \frac{p_{t_{i-1}^d w_{i-1}^d 1} + \gamma_1}{\sum_{c=0}^{1} p_{t_{i-1}^d w_{i-1}^d c} + \gamma_0 + \gamma_1} \times \frac{n_{w_i^d}^{w_{i-1}^d} + \delta_{w_i^d}}{\sum_{v=1}^{W} n_v^{w_{i-1}^d} + W\delta} \text{ and } t_i^d = t_{i-1}^d \tag{6.18}$$

6.4.2 Model Description of NNTM-2

The `NNTM-1` model in Section 6.4.1 has the ability to generate collocations where the words share the same topic. However, there is a shortcoming in the model. The model does not consider the contextual information when forming collocations. It means that it neither considers word nor topic context in generating collocations. This is exemplified by the following example. The `NNTM-1` model can generate collocations related to "green house", but it does not guarantee that this bigram will belong to "global warming" topic with a high probability. This limitation is handled in our second model `NNTM-2` which considers the context information when generating a bigram. So based on the context, if there is a possibility of bigram formation, then it will generate a bigram otherwise it will not generate for the same two words in different contexts. Thus we expect that this model is a more powerful generalization of our first model. However, the model has one drawback in that the number of parameters are more than our first proposed model. But we can notice that even such a slight variation in the topic model can bring an important benefit to the model, which will help improve the qualitative results tremendously. Our proposed `NNTM-2` model can do exactly what the `TNG` model proposed in [261], [175] can accomplish. One of the main differences is that the `TNG` model requires the user to pre-define the

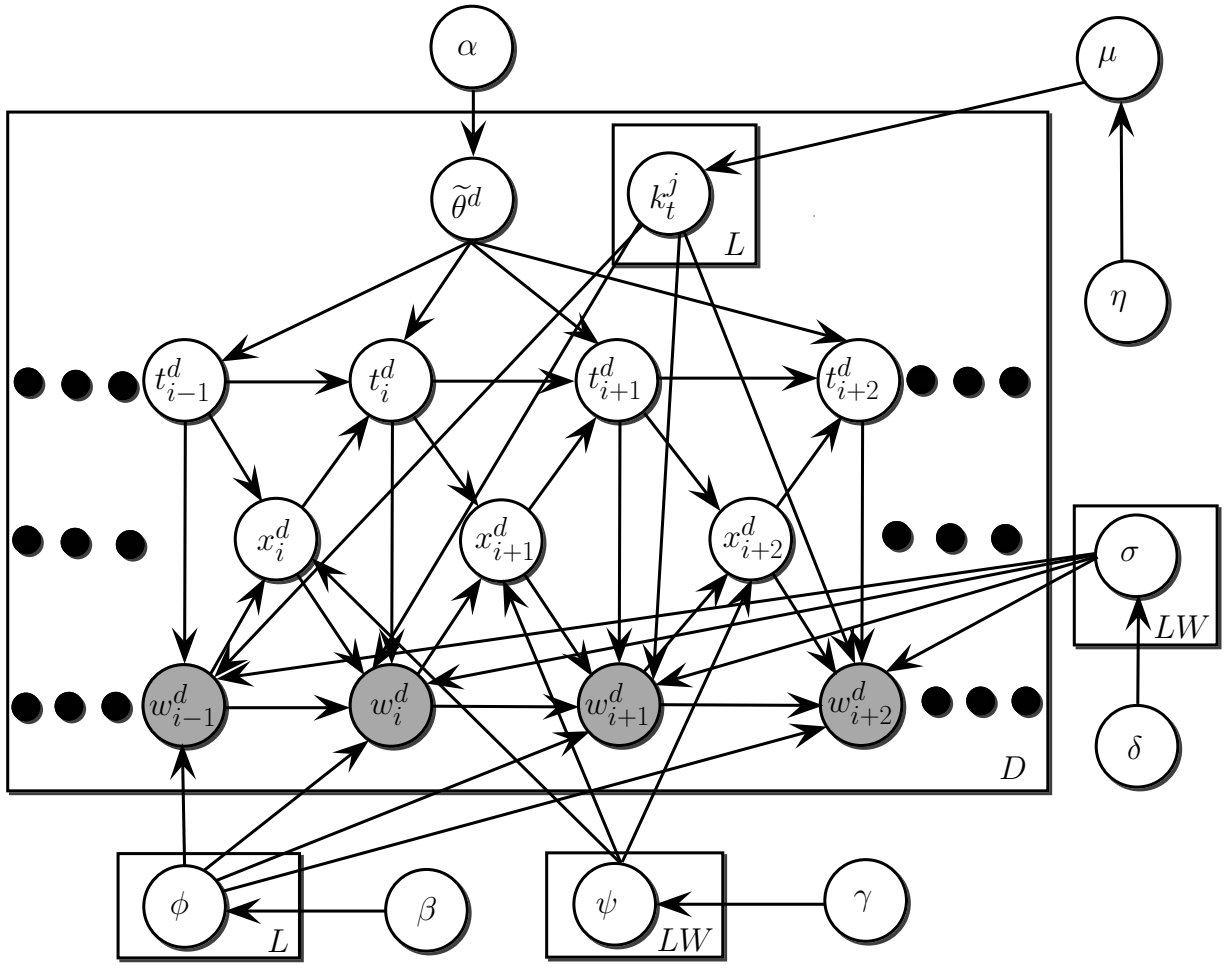


Figure 6.4: Our proposed NNTM-2 in Chinese Restaurant Franchise representation

number of latent topics, whereas our model does not. The TNG model does not give the same topic assignment to the same words in the bigram, and rather adopts an assumption where the topic of the head noun is chosen as the topic of the entire phrase.

The Generative Process

In this section, we present the generative process of the NNTM-2 model. The basic difference between our first proposed NNTM-1 and this model is that the generative process also considers the topic of the word.

1. Draw ϕ from $H(\beta)$
2. Draw μ from $\mathbf{GEM}(\eta)$
3. Draw $\mathbf{Discrete}(\sigma)$ from $\mathbf{Dirichlet}(\delta)$
4. Draw $\mathbf{Bernoulli}(\psi)$ from $\mathbf{Beta}(\gamma)$
5. For each document d
   (a) Draw $\widetilde{\theta}^d$ from α
   (b) Draw k_t^d from μ
   (c) For each word w_i^d at position i in the document d
      i. Draw t_i^d from $\widetilde{\theta}^d$ if $x_i^d = 0$ otherwise $t_i^d = t_{i-1}^d$
      ii. Draw x_i^d from $\mathbf{Bernoulli}(\psi_{t_{i-1}^d w_{i-1}^d})$
      iii. Draw w_i^d from $\phi_{k_{t_i^d}^d}$ if $x_i^d = 0$ else draw $\sigma_{t_i^d w_{i-1}^d}$

Posterior Inference

The posterior inference scheme of this model is similar to our previous model, `NNTM-1`, but with some variation. The difference is that in this model a word is generated not only by the previous word but also by its topic (of the current word), and the words share the same topic. This results in some change to the inference scheme when the status variable x_i^d between two words in sequence is 1. Therefore, the generation process can be described as:

1. {Partitioning} For each document $d \in D$
   (a) For each word at position $i \in d$ with $x_i^d = 0$

i. Draw the table index $t_i^d | t_1^d, t_2^d, \cdots, t_{i-1}^d, \alpha \sim \mathbf{CRP}(\alpha)$ otherwise set the table index $t_i^d = t_{i-1}^d$

(b) For each table $t \in d$

i. Draw the topic $k_t^d | k_1^1, k_2^1, \cdots, k_1^2, k_{t-1}^d, \gamma \sim \mathbf{CRP}(\gamma)$

2. {Generating} For each document $d \in D$

(a) For each word at position $i \in d$

i. Draw the bigram indicator variable $x_i^d \sim \psi_{t_{i-1}^d w_{i-1}^d}$

ii. **Draw a word** $w_i^d \sim \phi_{k_{t_i^d}^d}$ **when** $x_i^d = 0$ **else draw a word** $w_i^d \sim \sigma_{t_i^d w_{i-1}^d}$

The First Condition The first condition refers to $x_i^d = 0$. This sampling condition is the same as the one used in `NNTM-1` model.

The Second Condition We have stated earlier that when words in sequence form a collocation, then they share the same topic. With regard to the `CRF` scheme, it can be analyzed as two customers who are buddies come at the restaurant and choose the same table to sit.

In order to facilitate the sharing of the same topic for the words (customers), we know that one word (customer) is already assigned to a table in the restaurant. When a new word (customer) comes in, its assignment will be based on the previously assigned word. It means that if the previous word and the current word form a collocation then the current word is assigned to the same table as the previous word or both of them choose another table or open a new table altogether. This is where the dependence on the topic plays a role. In contrast, in our `NNTM-1` model, the customer who finds the buddy simply sits at the same table where the buddy was sitting as topic plays no role in the generation of words. Let us denote these two

words in sequence as $B^d_{i-1,i}$, which represents a bigram. These two words together can be considered as a single entity. This scheme can be expressed as given below.

The likelihood of $B^d_{i-1,i}$ for $t^d_{i-1,i} = t$ where t is an existing table, denoted as $f_k^{\neg B^d_{i-1,i}}(B^d_{i-1,i})$, is the conditional density of $B^d_{i-1,i}$ given all words in the topic k except the bigram $B^d_{i-1,i}$:

$$f_k^{\neg B^d_{i-1,i}}(B^d_{i-1,i}) = \frac{\int f(B^d_{i-1,i}|\phi_k) \prod_{d'i' \neq di, z_{d'i'}=k} f(B^{d'}_{i'i-1'}|\phi_k) h(\phi_k) \mathrm{d}\phi_k}{\int \prod_{d'i' \neq di, z_{d'i'}=k} f(B^{d'}_{i'i-1'}|\phi_k) h(\phi_k) \mathrm{d}\phi_k} \tag{6.19}$$

Just as in the first case, the equation can be simplified as follows:

$$f_k^{\neg B^d_{i-1,i}}(B^d_{i-1,i} = \vartheta) = \frac{n_{..k}^{\neg B^d_{i-1,i}, \vartheta} + \eta}{n_{..k}^{\neg B^d_{i-1,i}} + W\eta} \tag{6.20}$$

where $n_{..k}^{\neg B^d_{i-1,i}}$ is the number of words belonging to the topic k in the corpus excluding $B^d_{i-1,i}$. $n_{..k}^{\neg B^d_{i-1,i}, \vartheta}$ is the number of times the word ϑ is assigned with the topic k excluding $B^d_{i-1,i}$. The likelihood of $B^d_{i-1,i}$ for $t^d_{i-1,i} = \hat{t}$, where $\hat{t}$ is the new table being sampled, is written as:

$$P(B^d_{i-1,i}|t^d_i = \hat{t}, \mathbf{t}^{\neg di}, \mathbf{k}) = \sum_{k=1}^{L} \frac{m_{.k}}{m_{..} + \gamma} f_k^{\neg B^d_{i-1,i}}(B^d_{i-1,i}) + \frac{\gamma}{m_{..} + \gamma} f_{\hat{k}}^{\neg B^d_{i-1,i}}(B^d_{i-1,i}) \tag{6.21}$$

The probability of of a table is described as:

$$P(t_i^d = t|\mathbf{t}^{\neg di}, \mathbf{k}) \propto$$
$$\begin{cases} t_i^d = t_{i-1}^d & \text{if } w_{i-1}^d \text{ is already sitting there} \\ \alpha P(B_{i-1,i}^d | t_i^d = \hat{t}, \mathbf{t}^{\neg di}, \mathbf{k}) & \text{if } t = \hat{t} \end{cases} \tag{6.22}$$

Also, note that we have to decrease the customer count by 1 when a buddy customer leaves the table. So, Equation 6.19, can be written as, which can be regarded as a modified conditional density which will be used in the subsequent iterations of the sampler:

$$\hat{f}_k^{\neg w_i^d}(w_i^d = \vartheta) = \frac{(n_{..k}^{\neg w_i^d, \vartheta} - 1) + \eta}{(n_{..k}^{\neg w_i^d} - 1) + W\eta} \tag{6.23}$$

Sampling the Concatenation Indicator Variables

We present how to sample the values of the indicator variables. The idea is to compute the probabilities of how often two words consecutively occur in sequence. Then based on the probability value, the indicator variable is set to either 0 or 1. $f_k^{\neg w_i^d}(w_i^d = \vartheta)$ is defined in Equation 6.23. Let $n_{w_i^d w_{i-1}^d t_i^d}$ be the number of times the word w_i^d appears as a second word of a bigram with a previous word w_{i-1}^d and both words in the bigram are assigned to the same table t_i^d.

$$P(x_i^d = 0|\mathbf{x}^{\neg di}, \mathbf{w}, \mathbf{t}) = \frac{p_{t_{i-1}^d w_{i-1}^d 0} + \gamma_0}{\sum_{c=0}^{1} p_{t_{i-1}^d w_{i-1}^d c} + \gamma_0 + \gamma_1} \times f_k^{\neg \mathbf{w}_{dt}}(\mathbf{w}_{dt}) \tag{6.24}$$

$$P(x_i^d = 1|\mathbf{x}^{\neg di}, \mathbf{w}, \mathbf{t}) = \frac{p_{t_{i-1}^d w_{i-1}^d 1} + \gamma_1}{\sum_{c=0}^{1} p_{t_{i-1}^d w_{i-1}^d c} + \gamma_0 + \gamma_1} \times \frac{n_{w_i^d w_{i-1}^d t_i^d} + \delta_{w_i^d}}{\sum_{v=1}^{W} n_{v w_{i-1}^d t_i^d} + W\delta} \text{ and } t_i^d = t_{i-1}^d \tag{6.25}$$

6.5 Experiments and Results

6.5.1 Document Modeling Experiments

We describe our first quantitative analysis where we conduct document modeling. In particular, we use perplexity as the measure of the modeling quality. Perplexity has been widely used for document modeling in many works related to parametric and nonparametric topic modeling [247], [23]. It can be perceived of as the uncertainty in predicting a single word according to the model [231]. Perplexity is widely used in evaluating the quality of document modeling. To define perplexity, let M be the number of test documents. Let $\mathbf{w}_d$ is the vector of words in the document d. Perplexity for the test collection $\mathbf{D}_{test}$ is written as:

$$\text{Perplexity}(\mathbf{D}_{test}) = \exp\left(- \frac{\sum_{d=1}^{M} \log P(\mathbf{w}_d)}{\sum_{d=1}^{M} N^d} \right) \tag{6.26}$$

Low perplexity values represent better performance of the model. In the training phase, the word-topic and the document-topic matrices are learnt from the training data. This process finds out the unknown parameters of each multinomial distribution over words. During the testing phase, the count matrices learnt from the data can be used as the beginning point to which the counts from the word assignments to topic during the testing phase can be added. Detailed process of applying the learnt topic model on the testing dataset is explained in [45], [253].

Datasets

We use both small and large scale datasets for experimental evaluation. The statistics of the datasets are shown in Table 6.1.

Name	Number of documents	Total words	Words in vocabulary	Average document length
AQUAINT-1	1,033,461	221,751,420	80,201	429
NIPS	1,830	2,532,958	5349	2761
OHSUMED	233,448	20,116,637	18,722	166
Reuters	806,791	97,429,960	51,914	242

Table 6.1: Statistics of the datasets used in our document modeling experiment. "Total words" gives the total number of words in the entire collection. "Words in vocabulary" is the number of unique words in the collection. The last column gives the average number of words in a document.

Experimental Setup

We create five folds for each of these datasets and conduct five-fold cross validation. Each fold is created by randomly sampling 75% of the entire documents into the training set, and the rest into the testing set.

The comparative methods that we use in experiments consist of both parametric and nonparametric topic models. The parametric topic models are: `LDA` [23], `BTM` [252], `LDACOL` [87], `TNG` [261], and our recently proposed method `NTSeg` [117]. The nonparametric topic models are `HDP` [247], and our recently proposed model `NHDP` [116].

Parametric topic models require the hyperparameter values to be given explicitly. Based on the notations used in [23], for `LDA`, the hyperparameter values are $\alpha = \frac{50}{T}$, where T is the number of topics, and $\beta = 0.1$. In [260] the authors point out that in parametric topic models, the sensitivity to the variation of hyperparameter values is not significant. Based on the notations used in [261], for `BTM`, α is same as in the `LDA`

model, but we set as $\delta = 0.03$. The hyperparameter values for the LDACOL model are taken from the publicly available implementation[2]. The hyperparameter settings for the TNG model are the same as that used in its public implementation[3]. The values in NTSeg are already mentioned in [117]. In all the topic models Gibbs sampling has been used for doing the posterior inference.

As used in [247], we use the same hyperparameter settings in our HDP implementation. For our model, in order to make a fair comparison, we use the same hyperparameter settings that are commonly shared between our models and the HDP model. Some extra hyperparameter values that we assume are: $\gamma = 0.001$ and $\delta = 0.01$, for both of our models. Since our model inherits the exchangeability property of the HDP model, the sampling for the hyperparameters of the first and the second level prior will be the same as in [247]. Hyperparameter values for the NHDP model are the same as described in [116]. We have removed stopwords from the collection using the standard stopword list[4]. We have also performed stemming using Porter's stemmer. Note that we had tested the performance of the models using both stemmed and un-stemmed collections. We found that stemmed collections performed better than the unstemmed collections. All the experiments have been conducted by running the samplers for all the models five times for 1000 iterations in each fold. In order to present the final results, we computed the average for the five runs in each fold, and then computed the average of all the results obtained in the five folds.

Since the NIPS dataset is small, we have varied the number of topics from 10 to 100 in steps of 10 during the tuning process. In large collections, for example, OHSUMED, Reuters and AQUAINT-1, we have varied the number of topics from 50 to 200 in steps of 10 as larger collections tend to have larger number of topics [266].

[2]http://psiexp.ss.uci.edu/research/programs_data/toolbox.htm
[3]http://mallet.cs.umass.edu/
[4]http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

In parametric topic models, it is not straightforward to determine the number of topics that describes the collection. Researchers in the past have varied the number of topics, and have shown how varying the number of topics affects the final outcome. But rarely has it been shown about the number of topics that best describes the data. So one will wonder about the number of topics selected for a particular dataset. Simply selecting the best performing model from the results obtained by varying the number of topics is not a principled solution. Therefore in order to determine the best number of topics for comparison in our experiments, we adopt a tuning process. In the tuning process, in each fold, we first divide the training set into the development set which is 75% of the total number of documents in the training set, and the rest goes into the tuning set. We train the model using the development set and vary the number of topics. Then we compute the perplexity for each number of topics using the tuning set in each fold. Note that we also run the Gibbs sampler with 1000 iterations in each fold. Then we choose the best performing model through this procedure i.e. the model with the lowest average perplexity. We repeat five times and take the mode of the number of topics as the output of the tuning process. We then merge the development and the tuning sets together to get the training set where we train the model from the number of topics obtained using the tuning process. We test the model using the same number of topics on the testing set in each fold, by running five times and compute the average.

Results and Discussion

In Table 6.3, we present the results for various datasets and for different comparative methods. The first column lists the comparative models along with our two proposed models at the bottom of the table. Then we list out the perplexity results. From the results obtained in the AQUAINT-1 collection, we see that our `NNTM-1` model outperforms all the comparative methods. The performance of our `NNTM-2` model in this collection is rather poor, but still it is better than other comparative methods.

In contrast, in the NIPS collection, our NNTM-2 model performs better than the comparative methods. NNTM-1 model still shows good performance than other models. In OHSUMED collection, we can see that in the results obtained from our models is not very strong, but still they are better than other comparative methods. For the Reuters collection too, our models show good improvement over the comparative models.

Model	Perplexity			
	AQUAINT-1	NIPS	OHSUMED	Reuters
LDA	4599.48	834.45	2305.32	3490.12
BTM	4578.57	833.75	2229.96	3411.98
LDACOL	4501.44	831.45	2398.22	3298.76
TNG	4423.76	828.32	2315.72	3108.43
HDP	4322.32	825.43	2240.23	3192.54
NHDP	4495.32	820.56	2299.45	3102.53
NNTM-1	4201.33	815.32	2200.65	3099.44
NNTM-2	4222.54	803.98	2201.47	3002.29

Table 6.2: Results of documents modeling of various models. The lower the perplexity value, the better is the model.

From the results obtained in all the four datasets, we see that our models are the best performing ones. The HDP and the NHDP could not generalize well on the unseen collections. Parametric topic models also perform poorly in generalization on unseen data. This could be due to their good generalization ability on the tuning set, but low on the actually testing data. Our models could generalize well on the testing data showing their robustness against the comparative methods. One reason why our model works better than the state-of-the-art models is that our model applies better topic detection for words in the corpus. By giving the same topic assignment to the phrasal words in sequence, it can model text better than other models. Moreover, it is able to fit the data well using the hyperparameter sampling scheme, which automatically detects an appropriate number of topics based on the data characteristic. In contrast, for parametric topic models. The number of topics need to be determined. In Table 6.3 we present the results obtained from the tuning process for various parametric topic models. We list different parametric topic models

along with the tuned number of topics. One can note that if the collection is large, the tuned number of topics also tend to be large. This finding is consistent with [266] where the authors discovered that larger text collections tend to exhibit more number of topics. We have seen from the results that even tuning process may not obtain a good number of topics to fit the data well.

Model	Tuned Number of topics			
	AQUAINT-1	NIPS	OHSUMED	Reuters
LDA	160	70	190	120
BTM	190	50	140	80
LDACOL	170	60	140	110
TNG	160	60	180	120

Table 6.3: Table showing the number of topics obtained from the tuning process for different parametric topic models.

In addition, we also conduct a sensitivity analysis on the Dirichlet parameter η in the nonparametric topic models. We will see how this parameter plays a role in the overall perplexity of the model. Note that even parametric topic models have a topic Dirichlet parameter over topic distributions. We present the results of this analysis in Figure 6.5. From all the datasets we see that in the beginning when η is small, in most of the models, the perplexity is generally high. But when we arrive half-way between 0 and 1, we notice a gradual fall in the perplexity. Then the perplexity increases again. The results also point out that η has noticeable effects on the perplexity of the model when generalizing on the testing set.

As stated earlier, nonparametric topic models automatically detect the number of latent topics from the data characteristics. Now we will show the number of latent topics detected by different nonparametric topic models on different datasets. In order to determine the number of topics, we run the Gibbs sampler for five times in each fold on both the training and the testing sets. We find out the number of topics at the end of the 1000th iteration. We then take the mode from the list of five runs. We do the same for all the folds, and in the end we take the mode from the list of five folds and repeat the result.

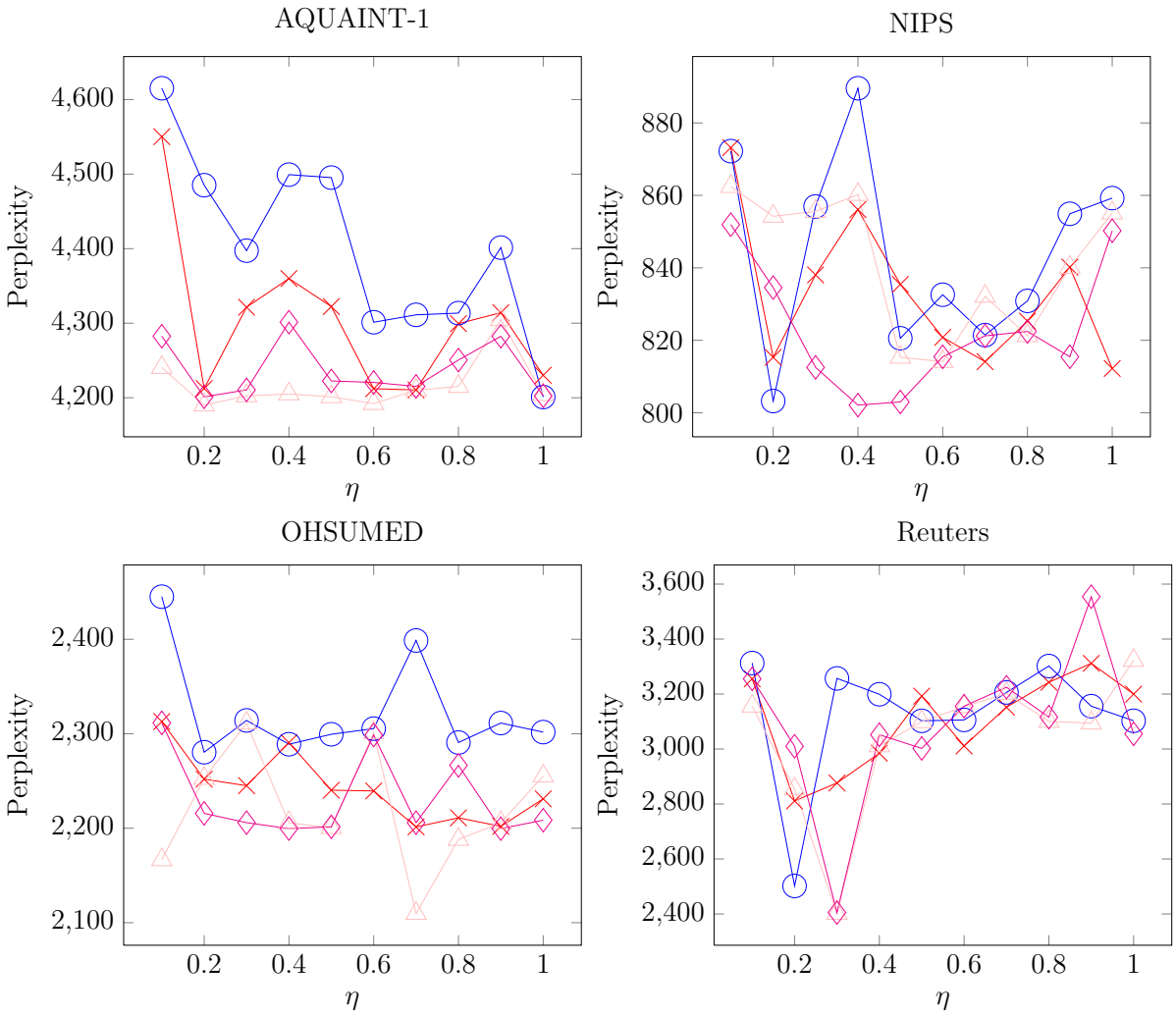


Figure 6.5: The effect of varying the topic Dirichlet parameter η on perplexity in the testing set for various nonparametric topic models. The marker with —×— is the `HDP` model. Similarly, —○— represents the `NHDP` model, —△— represents the `NNTM-1` model and —◇— represents the `NNTM-2` model.

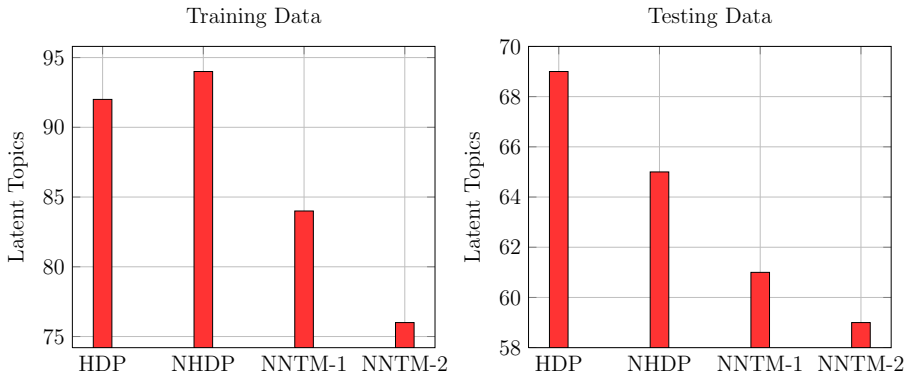


Figure 6.6: Number of topics detected by different nonparametric topic models on the training and the testing sets on NIPS collection.

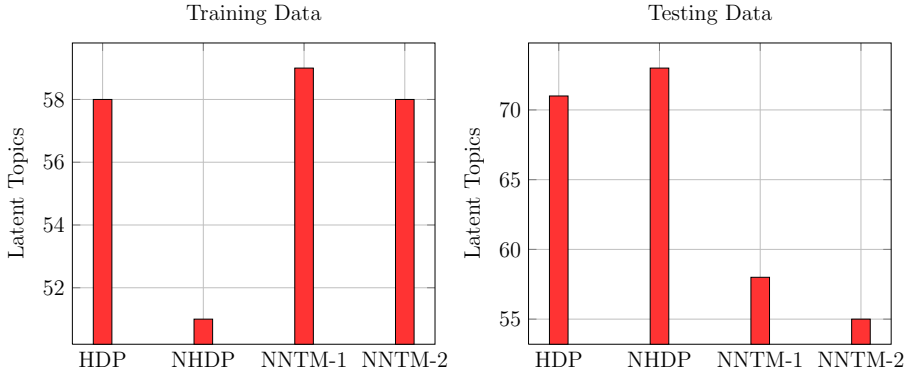


Figure 6.7: Number of topics computed by different nonparametric topic models on the training and the testing sets on OHSUMED collection.

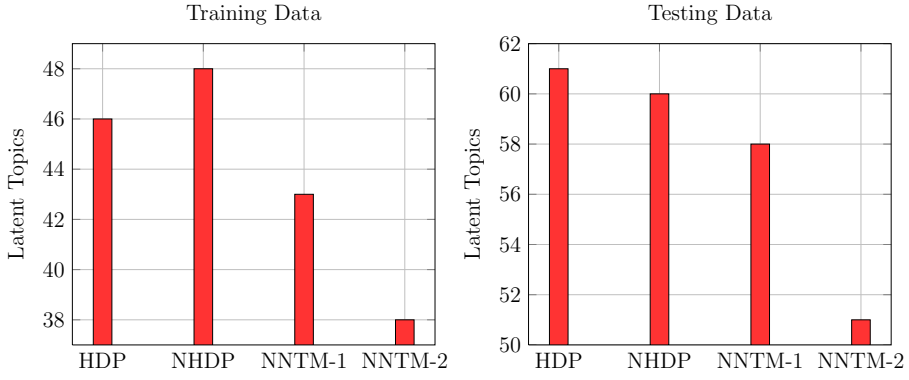


Figure 6.8: Number of topics computed by different nonparametric topic models on the training and the testing sets on Reuters collection.

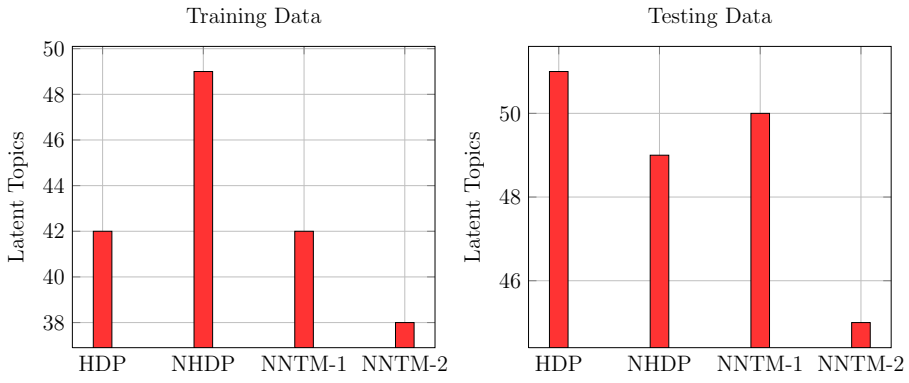


Figure 6.9: Number of topics computed by different nonparametric topic models on the training and the testing sets on AQUAINT-1 collection.

In Figures 6.6, 6.7, 6.8, 6.9, we present the number of latent topics automatically detected by the models for both the training and the testing sets. We notice that our `NNTM-2` model in all the datasets generates the least number of topics. This is primarily because most of the n-grams which share the same topic are clustered in one topic leading to the discovery of less number of topics. Our `NNTM-1` model also performs in a similar manner as the `NNTM-2` model. What is more noticeable is the uneven performance of the NHDP model which gives mixed set of results in all the datasets. HDP as usual will generate more number of topics as compared to our proposed `NNTM-1` and `NNTM-2` models.

One might notice that the number of latent topics generated by the nonparametric topic models is less than those obtained in the parametric topic models during the tuning process. We have indeed tested the parametric topic models with the number of topics generated by the nonparametric topic models, but we found that the performance of the parametric topic models were very weak when the number of topics was less. It is important to note that the values of the hyperparameters affect a lot in all the topic models that we have used for comparison. We in fact need some prior knowledge to find out a good value for such hyperparameters which in reality is very rare. Therefore, obtaining such results by vaguely fixing a hyperparameter value may lead to different results for different models.

6.5.2 Running Time Comparison

In this section, we present the running time comparisons for different nonparametric topic models. An obvious question that can arise after seeing the structure of our proposed graphical models in Figure 6.2 and Figure 6.4 is that our models are inherently quite complex which would lead to high computational overloads. But in reality this is not the case. We will illustrate the running time comparisons of different nonparametric topic models.

The extra overhead in our model lies in the Chinese Restaurant Franchise metaphor. Specifically, we need to search for friends inside the restaurant when a customer enters the restaurant. This searching time will be directly proportional to the number of customers already inside the restaurant and also the number of tables.

We show the running time comparison by computing the training time and testing time that each of the nonparametric topic models take. The running times are averaged over five folds in all the datasets. We present our results in Figures 6.10, 6.11, 6.12 and 6.13. In all the results we notice that the HDP model runs faster than the n-gram model which is obvious. Our models run faster in testing time in the NIPS collection. However, they also remain competitive to the HDP model in other datasets as well. We notice that in large document collections i.e. Reuters, AQUAINT-1 and OHSUMED, our models take more time in training, but the testing times is relatively less. Our objective is certainly not to show that our proposed n-gram models can run faster than the unigram based HDP model. The reason for fast computation time of HDP lies in its exchangeability assumption. But this assumption has been attacked for some time. Thus there is always a trade-off between time and improving upon the qualitative and quantitative results of the model. Our models have performed better than the HDP model, which bring out their importance in the field of text mining.

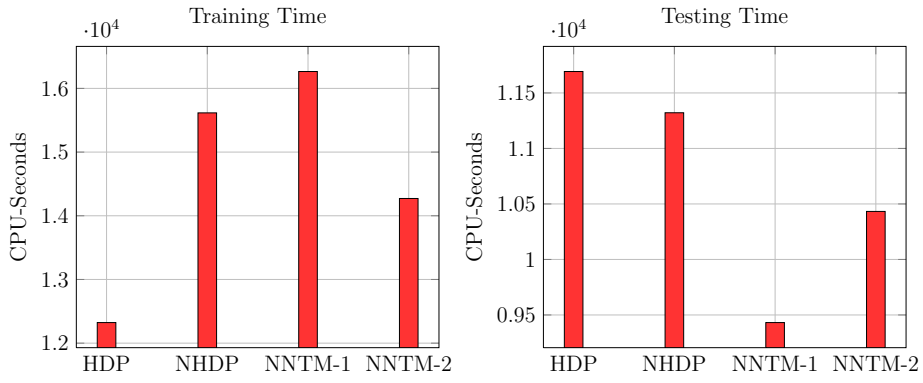


Figure 6.10: Training and testing time comparisons on NIPS collection for nonparametric topic models.

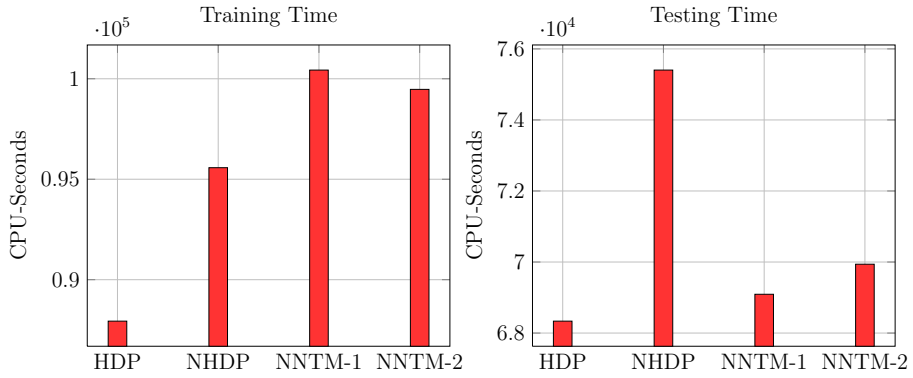


Figure 6.11: Training and testing time comparisons on OHSUMED collection for nonparametric topic models.

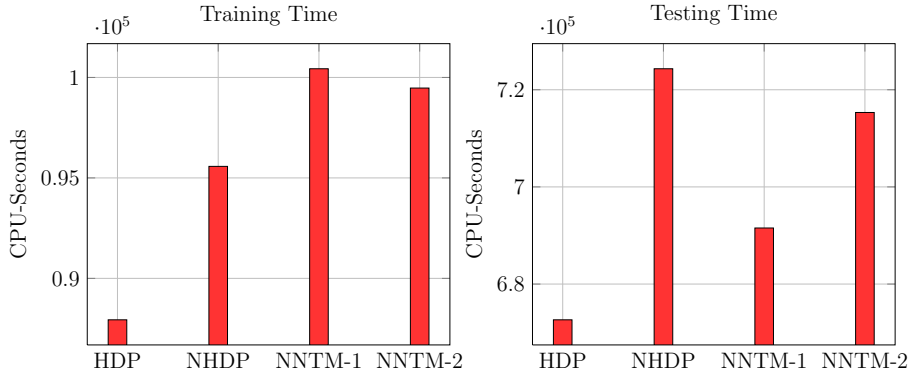


Figure 6.12: Training and testing time comparisons on Reuters collection for nonparametric topic models.

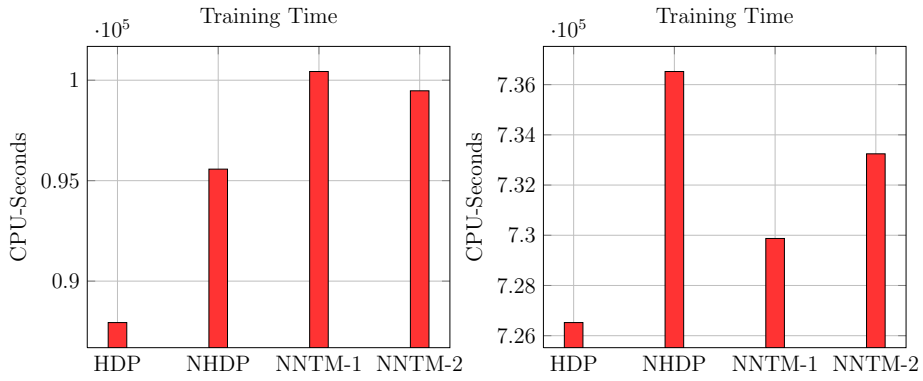


Figure 6.13: Training and tesing time comparisons on AQUAINT-1 collection for nonparametric topic models.

6.5.3 Qualitative Results

We conduct qualitative analysis where we show some high probability topical words obtained from our models and the comparative methods. The objective is to show that in comparison with the comparative methods our models generate much better topical words which seem to present much insightful topics as compared to the unigram based models.

In order to conduct qualitative analysis, we only consider nonparametric topic models for comparison. It is obvious that parametric class of n-gram model will likely generate similar or closely related interpretable topics than the unigram based models. The result presentation strategy is adopted from [261], as we also present unigrams and n-gram topical words separately as we are comparing with the unigram based HDP.

We present results from the AQUAINT collection in Tables 6.4 and 6.5. Results from the NIPS collection are presented in Tables 6.6 and 6.7. Similarly, results from OHSUMED are presented in Tables 6.8 and 6.9. Results from Reuters in Tables 6.10 and 6.11. Words in all the tables are ranked by their probabilities of occurrence in a latent topic. So word with the highest probability is at the top. In all the

results, we observe that n-gram models appear to produce more meaningful topical words than the unigram models. This is because the n-gram models lend more interpretability to the latent topics by generating n-gram words which appear to be more insightful than unigram words. In Table 6.5, we select words from technology related topics. When we see the words obtained from the HDP model, we note that we do not get much insight about the topics. For example, words such as "year", "new", "team" make the topic very ambiguous. Another weakness that we notice in the HDP model is that some of the words are not at all related to the topic, for example, "church". Therefore, we do not seem to get a sense about what the topic is actually referring to. In contrast, NHDP model generates slightly better unigrams than the HDP as most of the words are semantically related and will tend to co-occur together in documents. NHDP also generated n-grams which present better insight about the topic, however, words such as "index html" are not desirable ones, even a bigram "latin america" seems to be a misplaced one and does not fit in the topical content. In contrast, our two models have performed much better than the comparative methods. In NNTM-1, most of the unigrams generated are semantically closer to each other and describe about one theme. Also, the n-grams generated are all semantically related and talk about technology related theme. Also, one can note that by generating n-grams, we get rid of many ambiguities in the words present in the topics. For example, "web site', "computer device", etc, make more sense to the reader than simply generating unigrams as we can see that the n-gram words generated by our model are extremely useful in finding out about the theme of the topic. Similarly, NNTM-2 also generated better topic words than both NHDP and HDP. Also, in the NIPS dataset, our models generate better topical words than the comparative methods. For example, words generated by our NNTM-1 and NNTM-2 models appear more insightful than the comparative methods. Even the unigrams generated by our models appear more meaningful than the rest of the models. We notice similar results in the remaining two datasets too in rest of the tables. The experimental show that our models generate more fine grained topical words than rest of the models.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
year	test	internet sale	phone	web site	online	software package
game	computer	search engine	digit	cell phone	mail	multimedia game
music	year	create search engine	computers	high technology	program	rate children software
computer	project	internet user	technology	microsoft windows	computers	download free demo
train	modern	index html	information	computer technology	internet	web site
new	service	state department	web	computer device	technology	navigation system
team	software	computer software	mail	laptop equipment	software	computer vision
church	internet	computer bulletin	user	recognition software	information	computer software
transit	editor	latin america	online	large comfortable keyboard	site	computer technology
time	technology	talk real person	network	speech technology	system	microsoft windows

Table 6.4: Top ten n-grams from a topic related to "technology" obtained from each of the models from AQUAINT collection.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
dai	bosnia	nuclear submarine	belgrade	nato force	military	nato force
kosovo	solana	nato countries	troop	nato hold	war	black sea
mongolia	muslim	human suffering	yugoslavia	nato headquarters	kosovo	russian president
pipeline	war	serbian border	nato	nato naval command	nato	nato headquarters
stories	bosnia	checkpoint violent attack	albanian	nazi era	force	nato naval command
serb	arrest	albanian peasant	mirosev	russian royal navy	peace	nuclear submarine
mongolian	russian	russian modern submarine	kosovo	american vessel	mirosev	nato official
albanian	mexico	nato official	serb	navy official	belgrade	russian official
president	rescue	brussels headquarters	army	media report	albanian	russian navy spokesperson
nato	refuge	nato naval command	parliament	vladimir putin	bomb	media report

Table 6.5: Top ten n-grams from a topic related to "war" obtained from each of the models from AQUAINT collection.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
model	input	neural response	neurons	training data	train	neural network
image	training	neural population	spike	feature space	time	training set
network	network	training corpus	time	neural computation	synaptic	learning rule
neuron	output	content role	input	training step	network	firing rate
word	neuron	initial plan	network	input layer	input	action potential
sensors	synaptic	training instances	output	hidden unit	neuron	input neuron
train	time	neural network	neural	neural network	output	propagation filter
training	figure	hidden neuron	weight	neural net	spike	hidden neuron
algorithm	learning	training set	activation	hidden neuron	hidden	neural information
norm	data	neural systems	back	hidden unit	layer	training data

Table 6.6: Top ten n-grams from a topic related to "neural networks" obtained from each of the models from NIPS collection.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
function	signal	frequency division	hmm	speech enhancement	utterance	speech signal
sample	speaker	speech intelligibility	system	spectral subtraction	feature	frequency channel
time	hmm	neural tree	speech	test utterance	hmm	linear transformation
active	speech	complex tone	state	single microphone	speech	speech recognition
speech	time	speech parser	sound	human sound	phoneme	speech recognition system
input	frequency	atr ix	sources	spectral cues	output	training images
inputs	mlp	eig ts	recognition	noise suppression	channel	human sound
example	training	human sound	noise	time alignment	data	speech enhancement
set	recognition	ix el	speaker	speech enhancement	frequency	training case
linear	time	test utterance	time	training images	modeling	mit press

Table 6.7: Top ten n-grams from a topic related to "speech technology" obtained from each of the models from NIPS collection.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
patient	il	monoclonal antibody	cells	bone marrow	tissue	bone marrow
increase	specific	bone marrow	factor	monoclonal antibody	cells	cell clone
infect	human	cell line	effect	cell line	vitro	red blood cell
case	cell	human immunodeficiency virus	activity	peripheral blood	beta	monoclonal antibody
injuries	antibody	present study	vitro	red cell	human	tumor cell
cell	marrow	human specific	mrna	cell antigen	tissue	stimulating factor
arteries	bone	cell antigen	tissue	red cell	factor	cell antigen
receptors	line	cell activation	electron	cell clone	microscopy	cell activation
radical	virus	mm hg	collagen	cell activation	electron	result suggest
iga	human	results suggest	light	present study	skin	dna synthesis

Table 6.8: Top ten n-grams from a topic related to "cells" obtained from each of the models from OHSUMED collection.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
result	failure	renal transplant	liver	chronic liver disease	kidney	liver disease
control	kidney	plasma membrane	kidney	portal vein	liver	renal transplantation
liver	hepatic	liver biopsy	transplant	host disease	renal	portal vein
report	graft	hepatocellular carcinoma	patient	alcoholic liver disease	chronic	chronic liver disease
present	renal	host disease	hepatic	chronic active hepatitis	glomerular	liver biopsy
express	mm	mm hg	graft	plasma membrane	transplant	alcholic liver disease
flow	hg	body weight	portal	renal transplant	serum	hepatocellular carcinoma
antagonist	test	angiotensin ii	failure	hepatocellular carcinoma	graft	graft survival
vip	patient	mg days	renal	graft survival	creatinine	hepatic artery
allergy	liver	control subjects	chronic	liver disease	failure	alcholic liver disease

Table 6.9: Top ten n-grams from a topic related to "liver" obtained from each of the models from OHSUMED collection.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
market	bulgarian	andhra bank	market	india rbi	market	interest rate
percent	banker	european monetary union	foreign	term interest	share	india rbi
limit	commercial	complex financial product	growth	andhra bank	stock	stock exchange
crown	local	big deal	economy	interest rate	growth	dollar rate
pct	dealer	steel giant	insurance	land rate percent	fund	term interest rate
bank	boe	central bank buenos aires	industry	stock exchange	management	us dollar
share	anz	oil sale	development	belgian central bank	economy	daily interest rate
african	percent	russian trade system	sector	dollar rate	sector	government budget
space	yuan	floating rate	business	contact samir shah	financial	netprofit
bahrain	newspaper	german government	fund	financial institution	insurance	interest rate

Table 6.10: Top ten n-grams from a topic related to "finance" obtained from each of the models from Reuters collection.

HDP	NHDP		NNTM-1		NNTM-2	
	Unigrams	N-grams	Unigrams	N-grams	Unigrams	N-grams
report	year	oil product	oil	oil price	oil	crude oil
bank	japan	crude oil	trade	gulf war	cargo	domestic crude
win	iraq	new oil product	cargo	oil stock	barrel	oil product
pakistan	oil	january february	high	crude oil	gasoline	iraqi oil
oil	crude	saudi arabia	market	domestic crude	price	crude tanker
rate	demand	total product	price	iraq ambassador	fuel	oil storage
net	gasoline	crude export	fuel	oil product	crude	indian oil
french	saudi	gasoline distillation	tonne	indian oil	trader	chinese petroleum
launch	arabia	thousand barrel	crude	run oil company	limit	iraq ambassador
qatar	uae	oil import	week	world price	tonne	run oil company

Table 6.11: Top ten n-grams from a topic related to "oil" obtained from each of the models from Reuters collection.

The qualitative results can be summarized as follows:

1. Generally, words generated by the HDP model are inferior to that of the other models.

2. Our models are likely to generate better topical words than the other two nonparametric topic models.

3. Between our proposed models, it is difficult to make out which model is better than the other. This issue can be resolved through quantitative results that we have presented in Section 6.5.1 and 6.5.4 where we show the generalizing ability of the model on unseen documents.

4. N-gram words appear more meaningful than unigram words. This is because they give more insightful meaning to the reader. This has also been concurred in several other works in the past such as [166], [117].

6.5.4 Document Classification Experiments

In this section, we present document classification experimental results using topic models as conducted in [117]. We consider both parametric and nonparametric topic models in this experiment. We mainly consider closely related state-of-the-art topic models for comparison in our experiment. The purpose of this experiment is to show the performance of topic models in classifying an unseen document to its correct class. The comparison of topic model based on document classification with common text classification models as SVM has been investigated in [23], [223].

Datasets

We use two corpora in our experiments. One corpus is the 20 Newsgroups[5] which has been popularly used in several document classification tasks. Another corpus is OHSUMED-23 dataset[6] collection. Note that the OHSUMED-23 dataset is different from the OHSUMED dataset described in Section 6.5.1 that we have used in our document modeling experiments. We present more details about the corpora in Table 6.12.

Name	Number of Documents	Total words	Words in vocabulary	Average document length	Number of classes	Average number of documents in each class
20 Newsgroups	19,997	1,972,422	18,146	54	20	1000
OHSUMED	20,000	955,599	35,928	90	23	1016

Table 6.12: Statistics of the corpora used in our document classification experiment. Total words gives the total number of words in the entire collection. Words in the vocabulary is the number of unique words in the collection. The last column gives the average number of words in the document.

Classification Method

In the training phase, a topic model is learned for each class using the set of training documents in that class. In testing, we compute the likelihood of the testing document against each trained topic model for each class. The testing document is classified to the model that produces the highest likelihood. Note that this procedure is also used in [161], [117].

We measure the classification performance using precision, recall and F-measure. The meaning of precision for a class is the number of true positives divided by the total number of documents predicted to that class. Recall is defined as the number of true positives divided by the total number of elements that actually belong to that class in the gold standard. F-measure is the harmonic mean of precision and recall.

[5]http://qwone.com/~jason/20Newsgroups/
[6]http://disi.unitn.it/moschitti/corpora.htm

Experimental Setup

In the 20 Newsgroups corpus, we generated four datasets. The first dataset comprises of documents related to computer technology (the "comp" directory in the dataset). It is composed of several classes such as "graphics", "windows", "hardware", etc. Each of these classes consists of 1000 documents. The experimental setup is similar for the other three datasets, namely, "sci" (called Science Dataset), "politics" (called Politics Dataset), and "sports" (called Sports Dataset).

We create the training and the testing sets for each datasets. In the OHSUMED-23 collection, these splits are already provided. We trained a topic model in each class, and tested the performance using the testing set. We conducted this experiments using 5-fold and averaged the results from all five folds. We use the same values of the hyperparameters that we have used in our earlier experiments. We have described those hyperparameters in Section 6.5.1 including the hyperparameter sampling scheme.

We adopt similar text preprocessing strategy as adopted previously. Precisely we removed stopwords from the collections, stemmed the collections, and also removed some low frequency words i.e. words which occurred less than 8 in the entire collection.

Models	Precision	Recall	F-Measure
LDA	0.514	0.476	0.501
BTM	0.501	0.466	0.499
LDACOL	0.518	0.472	0.509
TNG	0.520	0.469	0.509
HDP	0.518	0.476	0.504
NHDP	0.496	0.491	0.483
NNTM-1	0.526	0.499	0.513
NNTM-2	0.501	0.438	0.509

Table 6.13: Computer Dataset

Models	Precision	Recall	F-Measure
LDA	0.416	0.392	0.392
BTM	0.401	0.376	0.376
LDACOL	0.405	0.322	0.394
TNG	0.411	0.339	0.399
HDP	0.416	0.401	0.405
NHDP	0.408	0.366	0.372
NNTM-1	0.415	0.405	0.405
NNTM-2	0.420	0.409	0.410

Table 6.14: Science Dataset

Models	Precision	Recall	F-Measure
LDA	0.412	0.401	0.376
BTM	0.415	0.401	0.398
LDACOL	0.416	0.402	0.389
TNG	0.411	0.399	0.399
HDP	0.418	0.401	0.405
NHDP	0.402	0.380	0.401
NNTM-1	0.416	0.401	0.402
NNTM-2	0.418	0.405	0.410

Table 6.15: Politics Dataset

Models	Precision	Recall	F-Measure
LDA	0.301	0.296	0.294
BTM	0.299	0.299	0.295
LDACOL	0.301	0.294	0.299
TNG	0.308	0.301	0.302
HDP	0.309	0.302	0.286
NHDP	0.302	0.296	0.292
NNTM-1	0.302	0.299	0.293
NNTM-2	0.303	0.301	0.303

Table 6.16: Sports Dataset

Models	Precision	Recall	F-Measure
LDA	0.514	0.468	0.499
BTM	0.512	0.466	0.501
LDACOL	0.512	0.472	0.472
TNG	0.516	0.473	0.486
HDP	0.516	0.482	0.492
NHDP	0.501	0.411	0.486
NNTM-1	0.514	0.476	0.511
NNTM-2	0.509	0.481	0.506

Table 6.17: Results obtained from the OHSUMED-23 corpus.

Result Analysis

We present our results in Tables 6.13, 6.14, 6.15 and 6.16. From the results, it is evident that our proposed models perform better than many parametric and non-parametric topic models in the task of classifying documents into its correct class. As far as our models are concerned, the results are mixed. In some cases, our first proposed model performs better than the latter. In the Computer dataset of the 20 Newsgroups corpus, which is shown in Table 6.13, we can see that our NNTM-1 model performs better than our NNTM-2 model. However, our NNTM-2 model is still competitive compared with other topic models. Nevertheless, our NNTM-1 model shows the best performance. The F-measure score of our model in Table 6.13 is statistically significant according to the sign test with p-value < 0.05 against each of the comparative models. In the Science dataset of the 20 Newsgroups corpus, shown in Table 6.14, both of our models show an improvement over other topic models. Although NNTM-1 performs at par with the HDP model, it still is competitive enough.

Our second model shows the best performance in this dataset. In the other two tables i.e. Tables 6.15 and 6.16, our NNTM-2 model shows the best performance. The improvements shown by our NNTM-2 models in these tables are statistically significant according to the sign test with p-value < 0.05 against each of the topic models.

We also present document classification results from the OHSUMED-23 dataset in Table 6.17. We can see from this result that NNTM-1 model has shown better performance than other topic models. However, our NNTM-2 model also performs better than other topic models in this dataset. The improvements shown by our NNTM-1 models in these tables is statistically significant according to the sign test with p-value < 0.05 against each of the topic models. Therefore, as stated in Section 6.4.2, our second model gives better document classification performance because it can capture topical words better than the NNTM-1 model.

In Figure 6.14, we present the effect of η on the classification performance. We have only considered the nonparametric topic models in this comparison because they are all closely related. We can notice from the figure that η has some impacts on the classification performance. The impact of η in the Science dataset is not that prominent though, whereas in the other datasets, η has a noticeable effect. In the Politics dataset, we can see that the HDP model almost matches the F-measure score of our model at $\eta = 0.5$. However, in most of the cases, our two proposed models fare well even when η is varied. In the Science dataset, HDP performs better than our NNTM-1 model for many values of η. One reason for the poor performance of our model in some cases might be due to the fact that there might not be many n-gram words in the documents. So in such a case, our model almost matches the HDP model as the potential power of our model is unleashed only when plenty of n-gram words exist in the document collection.

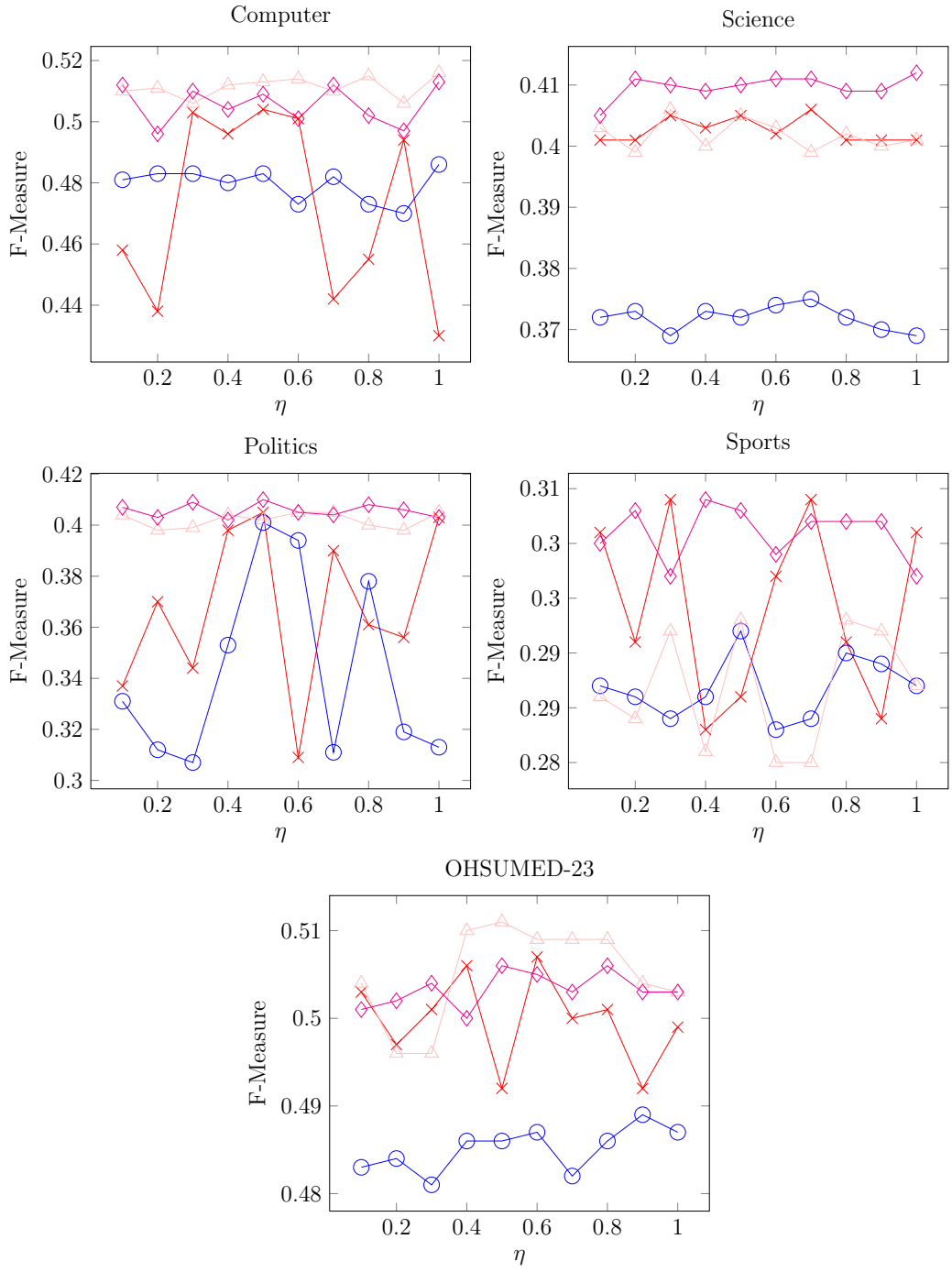


Figure 6.14: The effect of varying the topic Dirichlet parameter η on the classification performance for various nonparametric topic models. The marker with $-\times-$ is the `HDP` model. Similarly, $-\circ-$ represents the `NHDP` model, $-\triangle-$ represents the `NNTM-1` model and $-\diamond-$ represents the `NNTM-2` model.

6.6 Closing Remarks

In this chapter, we have proposed three nonparametric topic models to capture n-gram words in topics. Our model introduces a concatenation indicator binary variable in the model to capture n-gram topical words. We propose the Chinese Restaurant Franchise scheme with Buddy Customers to capture n-gram words in topics in nonparametric setting. One advantage of our model is that it does not require the user to specify the number of topics in advance which is needed in the parametric n-gram topic models. Our quantitative results show an improved performance in document modeling and document classification tasks. We also generate more meaningful topical words in topics generated by our model. We have shown that our model can be used in both small and large document collections.

Chapter Seven

Supervised Probabilistic Topic Models

Chapter Summary

One limitation of most existing topic models for document classification is that the topic model itself does not consider useful side-information, namely, class labels of documents. Topic models, which in turn consider the side-information, popularly known as supervised topic models, do not consider the word order structure in documents. In this chapter, we investigate a low-dimensional latent topic model for document classification. Class label information and word order structure are integrated into a supervised topic model enabling a more effective interaction among such information for solving document classification. We derive a collapsed Gibbs sampler for our model. Our experimental results suggest significant improvements over the state-of-the-art models.

7.1 The Case for the Supervised Topic Models With Word Order

Most existing topic models such as LDA are unsupervised probabilistic topic models which analyze a high dimensional term space and discover a low-dimensional topic space [23]. They have been employed for tackling text mining problems including document classification [117] and document retrieval [266], [261]. These models can achieve better performance via detecting the latent topic structure and establishing a relationship between the latent topic and the goal of the problem. One limitation of unsupervised topic models for document classification is that the topic model itself does not consider the class labels of documents. Another limitation of topic models is that they do not exploit the word order structure of the documents. Some works attempt to integrate the class label information into a topic model for solving document classification, for example, sLDA [21], mcLDA [257], and MedLDA [298]. These models have shown to improve document classification performance [299], [123]. However,

one common limitation of the above models is that they do not make use of the word order structure in text documents that could interact with the class label information for solving the document classification task.

Likewise, unsupervised topic models such as TNG and LDA have been used in developing document retrieval model [261], [266]. But they have not been explored for document retrieval learning which can be essentially cast into a learning-to-rank problem. Learning-to-rank models make use of available relevance judgment information of a document for a query in the training process. The task is then to predict a desired ordering of documents. Several learning-to-rank models have been introduced, but none of them consider the similarity between the document and the query under a low-dimensional topic space within the topic model itself.

In this chapter, we investigate a low-dimensional latent topic model for document classification. Class label information and word order structure are integrated into our supervised topic model enabling more effective interaction among such information for solving document classification. We derive a collapsed Gibbs sampler for our model.

7.2 Our Classification Model

7.2.1 Model Description

We propose a document classification model based on a latent topic model that integrates the class label information and the word order structure into the topic model itself. It enables interaction among such information for more effective modeling for document classification. There are two main components. One component measures the goodness of fit for document content similar to traditional topic models with

an extension of the consideration of the word order structure. The motivation for incorporating the word order structure is due to its ability to capture the semantic associations between the words in sequence [252], [117]. The second component deals with the prediction of class labels of documents. This component can be regarded as an extension of the Maximum Entropy Discrimination Latent Dirichlet Allocation (`MedLDA`) model [297], [298]. Essentially this component finds a regularized posterior distribution of the predictive function in a space defined by a set of expected margin constraints. This expected classifier is generalized from the maximum-margin constraints. One fundamental difference between `MedLDA` and our proposed model is that our model exploits the word order structure of a document. The design of the above two components leads to latent topic representation that is more discriminative and also advantageous for supervised document classification learning problem.

We show the graphical model in plate notation in Figure 7.1. The document content modeling component of our model is primarily a bigram supervised topic model which captures dependencies among the words in sequence. Each topic is characterized by a distribution of bigrams. A document is denoted by $d \in [1, \ldots, D]$ where D is the total number of documents. The training data is denoted by $T = \{\boldsymbol{w}^d, y^d\}_{d=1}^{D}$. We define $\boldsymbol{w}^d = \{w_i^d\}_{n=1}^{N^d}$ as words appearing in the document d. y^d is the class label which takes on one of the values $\mathtt{Y} = \{1, \ldots, M\}$. Word generation is defined by the conditional distribution $P(w_i^d | w_{i-1}^d, z_i^d)$. The goal of our model is to generate a latent topic representation that is suitable for classification task. We describe the generative procedure as follows:

1. Draw **Multinomial** distribution ϕ_{zw} from a **Dirichlet** prior β for each topic z and each word w.

2. For each document d

   (a) Draw a topic proportion θ^d for the document d from **Dirichlet** (α), where **Dirichlet** (α) is the Dirichlet distribution with the parameter α,

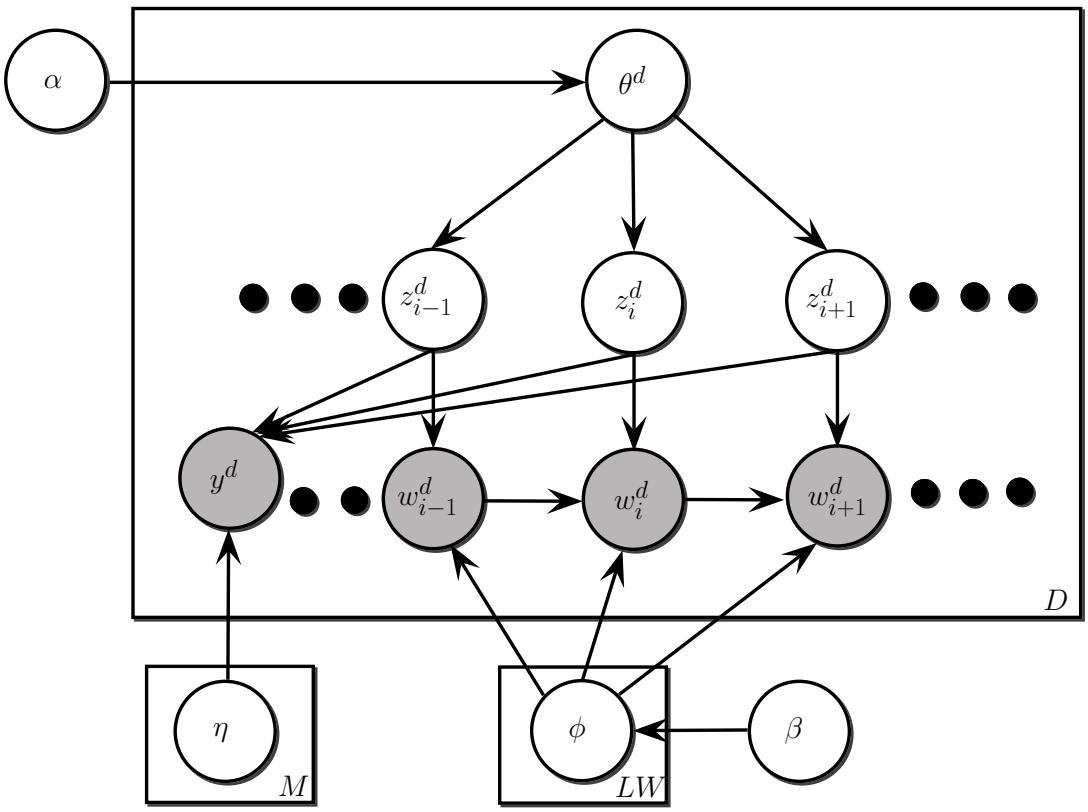


Figure 7.1: Graphical representation of our proposed document classification model.

(b) For each word w_i^d,

i. Draw a topic z_i^d from **Multinomial** (θ^d)

ii. Draw a word w_i^d from the distribution over words for the context defined by the topic z_i^d and the previous word w_{i-1}^d from **Multinomial** $(\phi_{w_{i-1}^d z_i^d})$

3. Draw the class label parameter η from **Normal** $(0, \eta_0)$, where η_0 is the hyper-parameter for η and is sampled M times,

4. Draw a class label $y^d|(\boldsymbol{z}^d, \eta)$

Let $\boldsymbol{W} = \{\boldsymbol{w}^d\}_{d=1}^D$ denote all documents in the training set. $\boldsymbol{Z} = \{\boldsymbol{z}^d\}_{d=1}^D$ are topic assignments to all the words. $\boldsymbol{\Theta} = \{\theta^d\}_{d=1}^D$ are topic distributions for all documents. $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_{kv}\}_{v,k=1}^{W,K}$ are word-topic distribution for the corpus. $\mathrm{KL}(P||P_0)$ is the Kullback-Leibler divergence from P to P_0. The joint distribution defined in the model is $P_0(\boldsymbol{\Theta}, \boldsymbol{\Phi}, \boldsymbol{Z}) = (\prod_{d=1}^D P(\theta^d|\alpha) \prod_n^W P(z_n^d|\theta^d)) \prod_{k=1}^K \prod_{v=1}^W P(\boldsymbol{\phi}_{kv}|\boldsymbol{\beta})$.

Similar to the supervised topic model which uses Gibbs sampling i.e. Gibbs Maximum-Margin Entropy Discrimination Latent Dirichlet Allocation (gMedLDA) in [123], our objective is to infer the joint distribution $P(\eta, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W})$, where η is a random variable representing the parameter of the classification model. Let $\boldsymbol{b}^d$ denote $\{b_{n,n+1}^d\}_{n=1}^{N^d-1}$, where $b_{n,n+1}^d$ denotes the words at the positions n and $n+1$ in the document d. $\boldsymbol{B} = \{\boldsymbol{b}^d\}_{d=1}^D$ is the word order information. In addition, we extend it to handle the word order structure. Precisely, the following discriminant function is defined as:

$$F(y, \eta, \boldsymbol{z}; \boldsymbol{b}^d) = \eta^\intercal \boldsymbol{f}(y; \overline{\boldsymbol{z}}^d) \tag{7.1}$$

where $\overline{\boldsymbol{z}}^d$ is a L dimensional vector with each element $\overline{z}_k^d = \frac{1}{N^d} \sum_{n=1}^{N^d} \mathbb{I}(z_n^d = k)$. $\mathbb{I}(.)$

is an indicator function which equals to 1 if the predicate holds else it is 0. $\boldsymbol{f}(y, \overline{\boldsymbol{z}}^d)$ is a MK-dimensional vector whose elements from $(y-1)$K to yK are $\overline{\boldsymbol{z}}_k^d$ and rest are all 0.

$$F(y; \boldsymbol{b}^d) = \mathbb{E}_{p(\eta, \boldsymbol{z}|\boldsymbol{b}^d)}[F(y, \eta, \boldsymbol{z}; \boldsymbol{b}^d)] \tag{7.2}$$

The prediction rule incorporating the word order structure in the classification task is:

$$\hat{y} = \underset{y}{\text{argmax}} F(y; \boldsymbol{b}^d) \tag{7.3}$$

Let C be a regularization constant, ξ^d be the slack variable and $l^d(y)$ be the loss function for the label y; all of which are positive. Following the idea in [123], the soft-margin for our model can be written as:

$$\begin{aligned} \underset{P(\eta, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \; & \text{KL}[P(\eta, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) || P_0(\eta, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] - \\ & \mathbb{E}_q[\log \text{P}(\boldsymbol{B}|\boldsymbol{Z}, \boldsymbol{\Phi})] + \\ & \frac{C}{D} \sum_d \text{argmax}_y(l^d(y)) - \mathbb{E}_P[\eta^\intercal(\boldsymbol{f}(y^d, \overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y, \overline{\boldsymbol{z}}^d))] \\ \text{subject to } & \mathbb{E}_P[\eta^\intercal(\boldsymbol{f}(y^d, \overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y, \overline{\boldsymbol{z}}^d))] \geq l^d(y) - \xi^d, \xi^d \geq 0, \forall d, \forall y, \end{aligned} \tag{7.4}$$

7.2.2 Posterior Inference

We use Collapsed Gibbs sampling for computing the posterior inference for our model in a similar manner in [123]. But we extend it to handle the word order structure in

the document. We factorize $P(\eta, \boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) = P(\eta)P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$. Then Equation 7.4 can be solved in two steps in alternate manner. The first step is to estimate $P(\eta)$ given $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$. In the second step, we need to estimate $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ given $P(\eta)$. The formulations are similar to that in [123] with a refinement for handling the word order structure.

Let $\boldsymbol{\Delta f}(y^d, \overline{\boldsymbol{z}}^d) = \boldsymbol{f}(y^d, \overline{\boldsymbol{z}}^d) - \boldsymbol{f}(y, \overline{\boldsymbol{z}}^d)$. The formulation for updating the posterior estimates is as follows:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \propto P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}, \boldsymbol{B}) e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{y^d} (\lambda_{y^d}^d)^* \boldsymbol{\Delta f}(y^d, \overline{\boldsymbol{z}}^d)} \tag{7.5}$$

where $\lambda_{y^d}^d$ is the Lagrange multiplier. The problem now is to efficiently draw samples from $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$. In order to simplify the integrals, we can take advantage of conjugate priors. We can integrate out the intermediate variables $\boldsymbol{\Theta}, \boldsymbol{\Phi}$ and build a Markov chain whose equilibrium distribution is the resulting marginal distribution $P(\boldsymbol{Z})$.

Let m_{zwv} be the number of times the word w is generated by the topic z when preceded by the word v. q_{dz} is the number of times a word is assigned to the topic z in the document d. We define $\boldsymbol{\kappa} = \sum_{d=1}^{D} \sum_{y^d} \lambda_{y^d}^d \boldsymbol{\Delta f}(y^d, \mathbb{E}[\overline{\boldsymbol{z}}^d])$, where $\boldsymbol{\kappa}$ is the mean of classifier parameters η. The element $\kappa_{y^d k}$ represents the contribution of the topic k in classifying a data point to the class y^d. The transition probability along with the maximum-margin constraint can be expressed as:

$$\begin{aligned} P(\boldsymbol{Z}|\boldsymbol{B}, \boldsymbol{Z}_{\neg n}, \alpha, \beta) = \Bigg(& \frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{K} \left(\alpha_z + q_{dz}\right) - 1} \times \\ & e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_y (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \Bigg) \times \frac{\beta_{w_i^d} + m_{z_i^d w_i^d w_{i-1}^d} - 1}{\sum_{v=1}^{W} (\beta_v + m_{z_i^d w_i^d v}) - 1} \end{aligned} \tag{7.6}$$

Note that all the counts used above exclude the current case i.e., the word being

visited during sampling.

Our prediction framework also follows similar strategy as in [123], but we consider the notion of word order. It would not be difficult to derive the prediction formulae based on the formulations that we have already presented in the above sections.

7.3 Document Classification Experiments

7.3.1 Experimental Setup

We conduct extensive experiments on document classification using existing benchmark test collections. We also compare with many related comparative methods. In addition, we carry out qualitative analysis showing how our model generates better topical words. In all our experiments for topic models, we run the sampler for 1000 iterations. We have also removed stopwords[1] and performed stemming using Porter's stemmer[2]. Five-fold cross validation is used as in [298]. In each fold, the macro-average across the classes is computed. Each model is run for five times. We take the average of the results obtained for all the runs and in all the folds.

We use two popular datasets, namely, 20 Newsgroups dataset[3] and OHSUMED-23 dataset[4]. The 20 Newsgroups dataset contains 1000 documents in each of the 20 classes. OHSUMED-23 comprises 23 classes. As adopted in [126], we used the first 20,000 documents divided into 10,000 as training documents and 10,000 for testing. In 20 Newsgroups, our training set comprised 75% of the total documents, and the rest as testing. We use Precision, Recall and F-measure to measure the

[1] http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

[2] We also tested the models without performing stemming. We found that stemmed collections fared better.

[3] http://qwone.com/~jason/20Newsgroups/

[4] http://disi.unitn.it/moschitti/corpora.htm

classification performance. The definitions for these metrics in classification task can be found in [117]. We solve multiclass classification problem by decomposing into binary classification problems in each class. We adopted a tuning process where we split the training set into 75% development set and the rest as tuning set. We used the development set to train the model, and we varied the number of topics from 10 to 100 in steps of 10 as in [117]. We performed this procedure in each fold and computed the average F-measure. The number of topics which produced the best F-measure is the output of the tuning process. Then we used the original training set (i.e. the development set and the tuning set) to train the models using the number of topics obtained from the tuning process. In order to present the number of topics in Table 7.1, we took the mode of the number of topics among all the five folds. We set the loss function ($l^d(y)$) to a constant function 16 just as in [123]. For simplicity, we assume all symmetric Dirichlet priors, and we set the value of β to 0.01. As experimented in [260], we also found not much variation in results with different hyperparameter values. Hyperparameter values of the other topic models (supervised and unsupervised) are the same as used in their respective works and their available publicly shared implementations. In [123], the authors conducted extensive experimentation to find the best C value. We use the same C value for fair comparison.

We chose a wide range of comparative methods as follows.

1. `gMedLDA` [299] denoted as `gMedLDA` in the results.
2. Variational Maximum-Margin Entropy Discrimination Latent Dirichlet Allocation (`vMedLDA`) [297] denoted as `vMedLDA`.
3. `sLDA` denoted as `sLDA` [21].
4. `DiscLDA` [147] denoted as `DiscLDA`
5. `LDA` [23].

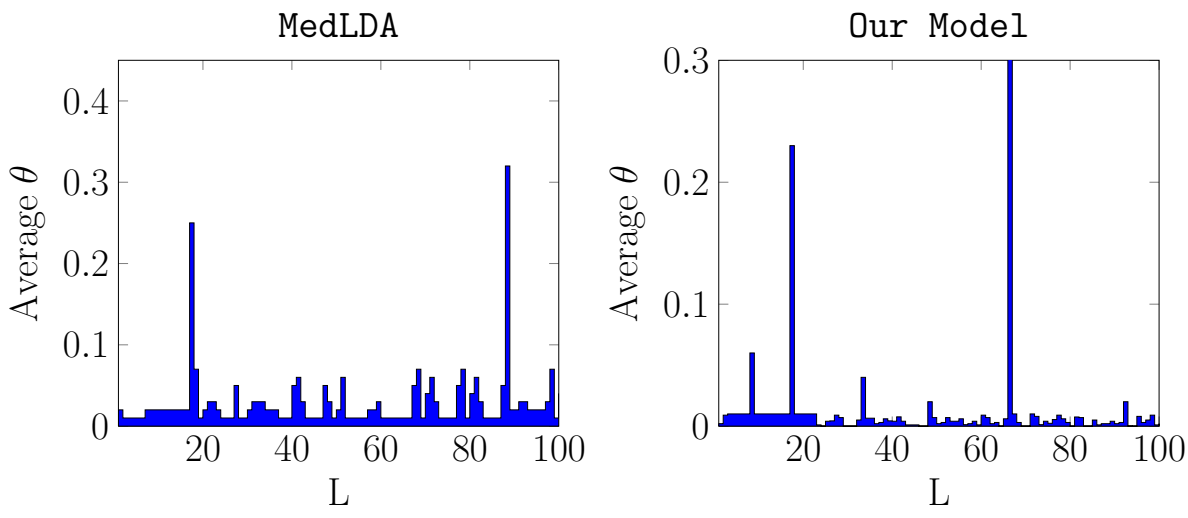


Figure 7.2: Per-class distribution over topics in *comp.graphics* class of 20 Newsgroups dataset.

6. We use `LDA+SVM` in the same way as described in [298].

7. Bigram Topic Model `BTM` [252].

8. Following procedure as adopted for `LDA+SVM`, we do the same for `BTM+SVM`.

9. `LDA`-Collocation model (`LDACOL`) [87].

10. `LDACOL+SVM`.

11. Topical N-gram (`TNG`) [261].

12. `TNG+SVM`, [126].

13. Our recently proposed model `NTSeg` [117].

14. `NTSeg+SVM`.

15. `SVM`. The features for linear `SVM` are same as that in [299].

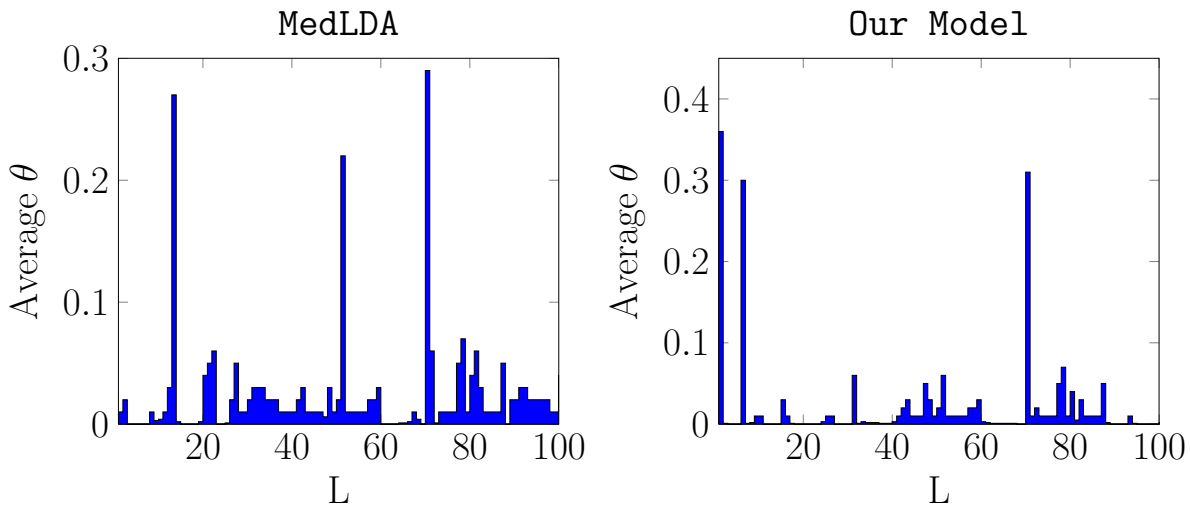


Figure 7.3: Per-class distribution over topics in Class 5 of OHSUMED-23 dataset.

Dataset	20 Newsgroups															
Models	Our Model	gMedLDA	vMedLDA	sLDA	DiscLDA	LDA	LDA+SVM	BTM	BTM+SVM	LDACOL	LDACOL+SVM	TNG	TNG+SVM	NTSeg	NTSeg+SVM	SVM
Topics	80	50	30	60	70	50	50	80	80	60	70	70	60	60	60	
Pre	0.945	0.869	0.865	0.805	0.756	0.859	0.835	0.877	0.835	0.843	0.845	0.845	0.832	0.766	0.869	0.825
Rec	0.916	0.869	0.865	0.812	0.780	0.859	0.920	0.848	0.920	0.914	0.932	0.932	0.866	0.905	0.845	0.910
F1	0.930	0.868	0.857	0.799	0.741	0.858	0.862	0.862	0.862	0.862	0.864	0.865	0.861	0.866	0.858	0.852
Dataset	OHSUMED-23															
Models	Our Model	gMedLDA	vMedLDA	sLDA	DiscLDA	LDA	LDA+SVM	BTM	BTM+SVM	LDACOL	LDACOL+SVM	TNG	TNG+SVM	NTSeg	NTSeg+SVM	SVM
Topics	70	40	60	60	70	40	40	60	40	50	50	60	60	40	40	
Pre	0.496	0.456	0.489	0.456	0.402	0.465	0.463	0.422	0.545	0.534	0.534	0.432	0.442	0.531	0.522	0.483
Rec	0.915	0.814	0.821	0.802	0.735	0.801	0.798	0.767	0.776	0.742	0.744	0.711	0.710	0.779	0.765	0.903
F1	0.643	0.633	0.629	0.620	0.587	0.626	0.631	0.610	0.622	0.630	0.625	0.623	0.620	0.634	0.623	0.630

Table 7.1: Classification results. Pre stands for Precision, Rec for Recall, and F1 for F-Measure. The row "Topic" lists out the number of topics which we have obtained from the tuning process.

7.3.2 Quantitative Results

We present our main classification results in Table 7.1. In each cell in the table, we first present the name of the model, then present the number of topics obtained from the tuning process, and in the last cell the first number is Precision (denoted as Pre), followed by Recall (denoted as Rec) and then F-Measure (denoted as F1). We observe that our model has outperformed all the comparative methods. In both datasets, our F-measure results are statistically significant based on the sign test with a p-value < 0.05 against each of the comparative methods. By maintaining the word order and considering an extra side-information helps in improving classification results to a great extent. Since we are capturing the inherent word order semantics

in the document, just like other structured unsupervised topic models, we obtained state-of-the-art improvement over the comparative methods. In Table 7.2, we study

20 Newsgroups										
Models	10	20	30	40	50	60	70	80	90	100
Our Model	0.783	0.843	0.899	0.914	0.922	0.921	0.925	0.930	0.927	0.924
gMedLDA	0.424	0.694	0.826	0.859	0.868	0.866	0.858	0.869	0.852	0.850
vMedLDA	0.245	0.667	0.857	0.852	0.843	0.831	0.818	0.802	0.789	0.777
sLDA	0.301	0.505	0.578	0.789	0.800	0.799	0.766	0.698	0.653	0.493
DiscLDA	0.245	0.452	0.643	0.654	0.701	0.743	0.741	0.699	0.636	0.545
LDA	0.410	0.683	0.816	0.849	0.858	0.856	0.848	0.859	0.842	0.840
LDA+SVM	0.752	0.802	0.827	0.837	0.862	0.844	0.850	0.851	0.842	0.839
BTM	0.715	0.775	0.831	0.846	0.854	0.853	0.857	0.862	0.859	0.856
BTM+SVM	0.552	0.602	0.807	0.816	0.849	0.857	0.863	0.862	0.856	0.787
LDACOL	0.601	0.633	0.701	0.699	0.843	0.862	0.854	0.833	0.765	0.799
LDACOL+SVM	0.545	0.601	0.812	0.824	0.834	0.859	0.864	0.851	0.855	0.799
TNG	0.552	0.615	0.803	0.819	0.831	0.857	0.865	0.835	0.803	0.772
TNG+SVM	0.556	0.612	0.816	0.824	0.835	0.861	0.866	0.859	0.862	0.845
NTSeg	0.601	0.612	0.654	0.670	0.840	0.866	0.845	0.756	0.722	0.626
NTSeg+SVM	0.646	0.640	0.745	0.801	0.855	0.858	0.806	0.703	0.603	0.515
OHSUMED-23										
Models	10	20	30	40	50	60	70	80	90	100
Our Model	0.597	0.600	0.605	0.644	0.642	0.642	0.643	0.643	0.644	0.642
gMedLDA	0.543	0.555	0.580	0.633	0.621	0.613	0.588	0.590	0.574	0.534
vMedLDA	0.542	0.556	0.552	0.558	0.585	0.629	0.632	0.611	0.589	0.534
sLDA	0.543	0.545	0.512	0.555	0.534	0.620	0.613	0.603	0.603	0.585
DiscLDA	0.503	0.502	0.512	0.507	0.532	0.611	0.587	0.575	0.545	0.543
LDA	0.545	0.593	0.565	0.626	0.611	0.615	0.601	0.599	0.546	0.600
LDA+SVM	0.542	0.585	0.556	0.631	0.605	0.610	0.587	0.585	0.535	0.598
BTM	0.546	0.590	0.594	0.630	0.630	0.610	0.576	0.554	0.523	0.554
BTM+SVM	0.511	0.545	0.578	0.622	0.625	0.613	0.572	0.553	0.526	0.524
LDACOL	0.513	0.575	0.565	0.631	0.630	0.601	0.569	0.523	0.514	0.515
LDACOL+SVM	0.499	0.504	0.560	0.631	0.625	0.601	0.567	0.522	0.512	0.531
TNG	0.523	0.572	0.554	0.610	0.625	0.623	0.621	0.524	0.552	0.520
TNG+SVM	0.524	0.573	0.550	0.606	0.622	0.620	0.622	0.527	0.543	0.519
NTSeg	0.524	0.579	0.560	0.634	0.629	0.598	0.554	0.515	0.512	0.555
NTSeg+SVM	0.516	0.560	0.554	0.623	0.612	0.584	0.498	0.515	0.513	0.525

Table 7.2: The effect of the number of topics on document classification measured by F-measure.

the effect of document classification performance as measured by F-measure when we vary the number of topics from 10 to 100 for parametric topic models. As we begin from $L = 10$ in the 20 Newsgroups dataset, we see that our model does not perform very well in the beginning. Nevertheless, it still outperforms other topic models. Our model performs very well after $L \geq 30$. Similarly, in the OHSUMED-23 dataset, our model also does not perform well for $L \leq 30$. Nevertheless, it still outperforms other topic models. Then it gains good improvement as we increase the number of latent

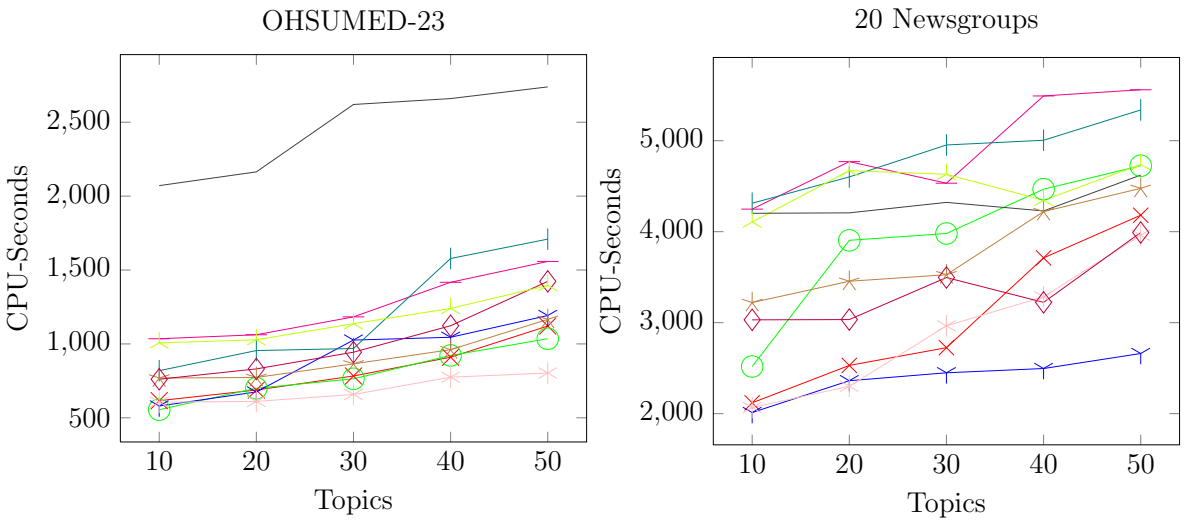


Figure 7.4: A plot showing the prediction time in CPU-Seconds on the classification datasets for various models. The model in ─×─ is our proposed model, `gMedLDA` is shown in ─○─, `vMedLDA` in ─+─, `sLDA` in ──, `DiscLDA` in ─┼─, `LDA` in ─✳─, `LDACOL` in ─◇─, `TNG` in ─Y─ and `NTSeg` in ── on the right.

topics. It can be observed that the unsupervised n-gram topic models' performance cannot be discarded. For example, the recently proposed method `NTSeg` has done well when compared to other unsupervised topic model in the 20 Newsgroups dataset. Similar pattern is observed in the OHSUMED-23 dataset. It suggests that word order plays an important role in influencing document classification performance. Our model can outperform the other comparative methods because it inherits the advantages of both n-gram unsupervised topic models. Note that as exemplified in [117] and many other works which follow word order, computational complexity of the models that follow word order is generally higher than those of their bag-of-words counterparts. Nevertheless, word order structure models have shown superior performance than the bag-of-words models [117]. Several attempts have been made recently to speed up the inference procedures for both supervised and unsupervised topic models such as [300], [301], [208].

Figure 7.4, presents the total prediction time for all the topic models on the testing set of two document collections. We follow this computation strategy from

[123]. All models are implemented in C programming language, and were run on a machine with 16GB of primary memory with dual core Intel having a clock speed of 3.33Ghz. Since the number of topics has direct impact on the prediction time, for fair comparison we depict different number of topics, and then compare the prediction time on the testing set. The CPU seconds are averaged over five runs in each fold, and then averaged over all the five folds. One can note from the results that although our model follows word order, it almost matches the testing times of the unigram based supervised topic models most of the time. Thus it shows that maintaining word order does not have an adverse effect on the prediction time of our model.

7.3.3 Qualitative Results

We qualitatively compare our model with some related n-gram and supervised topic models, including BTM [252], LDACOL [87], TNG [261], PDLDA [166], NTSeg [117], and MedLDA [298]. We present top five most representative words from a topic describing semantically similar theme from each model. The criterion for choosing the words from each topic are as follows. The sampler of each model was run for 1000 iterations. Then the perplexity at each latent topic value L was observed. We chose the words from that L that gave the lowest perplexity at the end of 1000 iterations. We chose the documents from *comp.graphics* class for qualitative experiments as adopted in [298]. From the results shown in Table 7.3, we can make two observations. First,

BTM	LDACOL	TNG	PDLDA	NTSeg	MedLDA	Our Model
compgraph path	xref	vga mode	excel digit	surface normal	path	bitmap draw
xref compgraph	compgraph	routine	remove	orient message id	routing	video memory
system distribution	compgraph path	pixmap	public domain	corporate	college	simple routing
problem solving	mark	public domain	draw line	copyright	date	color gif
fast purpose	compgraph subject	credit	message id	make group	sender	package zip

Table 7.3: Top five probable words from a topic from *comp.graphics* class of 20 Newsgroups dataset.

our model generates more fine grained topical words as compared to other topic models. Second, our model generates more interpretable latent topics as compared

to other topics. Words such as "video memory" etc, appear more appealing to a reader as compared to other models which in some instances make no sense. Other models rather generate ambiguous n-grams or they generate unigrams which do not offer much understanding to the user. Phrase discovery topic model, PDLDA, fails to capture better phrases primarily because it assumes that documents contain many phrases, and it tends to underperform on small documents which have less phrasal terms as those in the *comp.graphics* dataset.

In Figures 7.2 and 7.3, we illustrate the discriminative power of our model in comparison to the MedLDA model. In order to construct these plots we follow similar procedure as described in [298]. We can see from the plots that our model leads to sharper, more sparse and faster decaying per-class distributions over topics than the MedLDA model using Gibbs sampling. The plots show that our model has better discriminative power to differentiate between topics than its bag-of-words counterpart.

7.4 Closing Remarks

We have presented supervised topic models which maintain word order structure in the document. We have proposed a bigram supervised topic model with maximum-margin framework, and compare the performance of the model with several existing comparative methods. We saw through empirical analysis that our model outperforms many comparative methods. We can see from the experimental results that word order has helped improve document classification results considerably.

Chapter Eight

Document Retrieval Learning Models

Chapter Summary

In this chapter, we will present topic models for document retrieval learning problem, which can be essentially cast into a learning-to-rank problem. Learning-to-rank models make use of available relevance judgment information of a document for a query in the training process. The task is then to predict a desired ordering of documents. Several learning-to-rank models have been introduced, but none of them consider the similarity between the document and the query under a low-dimensional topic space within the topic model itself. We thus introduce a topic similarity feature in the learning-to-rank framework in a unified model.

8.1 The Case for the Supervised Document Retrieval Learning Topic Model

Unsupervised topic models such as TNG and LDA have been used in developing document retrieval model [261], [266]. But they have not been explored for document retrieval learning which can be essentially cast into a learning-to-rank problem. Learning-to-rank models make use of available relevance judgment information of a document for a query in the training process. The task is then to predict a desired ordering of documents. Several learning-to-rank models have been introduced, but none of them consider the similarity between the document and the query under a low-dimensional topic space within the topic model itself.

Mostly learning-to-rank models have considered two types of document features. One of them consists of the high-level features, and the other comprises of the low-level features as described in the Background section (Section 3). What the learning-to-rank models do not consider is the latent topic feature. The motivation is that latent topic models have shown to improve document retrieval performance in a

traditional setting [261], [266], but have not been investigated under a learning-to-rank setting or document retrieval learning setting. We hope that if a learning-to-rank framework considers the latent topic information during learning a ranking function, there will be significant improvements over the state-of-the-art models.

Our contribution is that we propose a new supervised topic model for document retrieval learning which can be regarded as a pointwise model for tackling learning-to-rank task. Available relevance assessments and word order structure are integrated into the topic model itself. We jointly model the similarity between the query and the document under a low-dimensional topic space in a maximum-margin framework. We conduct extensive experiments on several publicly available benchmark datasets, and show that our model improves upon the state-of-the-art models. One major difference between our model and existing learning-to-rank models is that existing learning-to-rank models do not consider latent topic information in the learning framework.

8.2 Model Description

We investigate a supervised low-dimensional latent topic model for document retrieval learning. Suppose that some relevance assessments of documents for some queries are available for training. Our goal is to learn a model that can predict the relevance of an unseen test query-document pair, and rank the documents based on the predicted relevance score. This problem setting is similar to the pointwise learning-to-rank problem. Manual relevance assessments can be modeled as a response variable in our topic model. In addition, the word order structure of the text content is also considered. The graphical model of our proposed document retrieval learning model is depicted in Figure 8.1. This model can be regarded as a basic document retrieval learning topic model, and upon which we will build on further to incorporate word order information. Based on the past empirical evaluations, we ex-

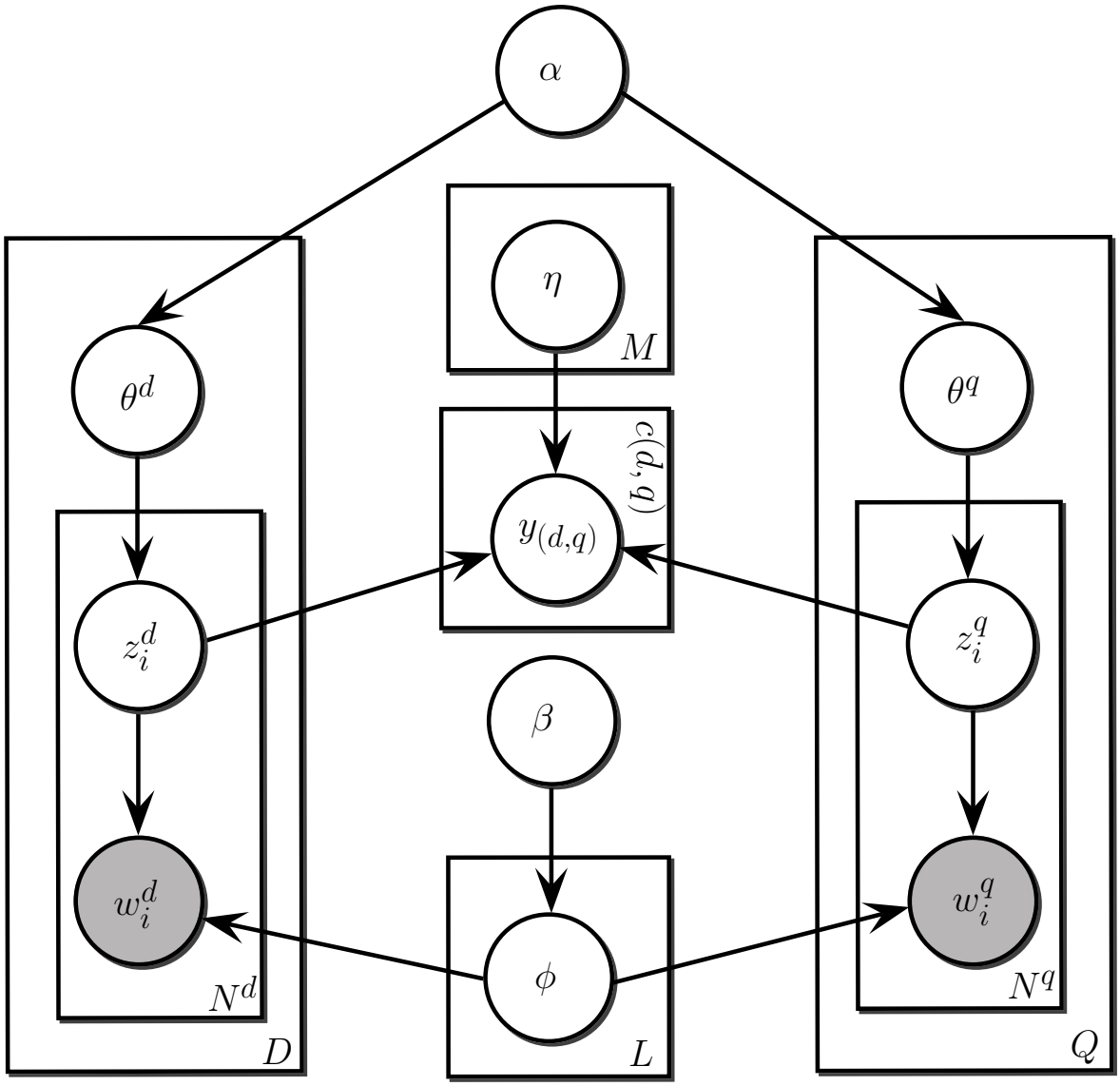


Figure 8.1: The graphical model of the our document retrieval learning model where order of words in not maintained. This graphical model is the simplest illustration of our model. The model comprises of two plates where one plate has the document information and the other is the query plate. The latent topic information of the document and the query are connected by the relevance label which is supplied as a response variable.

pect this topic model not to give better results because it loses important contextual information that is inherent in the document.

Similar to our proposed document classification model described in Chapter 7, there are two main components in our document retrieval learning model. One component is a topic model which measures the goodness of fit of the text content of documents and queries. Queries are modeled as short documents in a similar manner as in [271]. The second component deals with the relevance prediction within a maximum-margin framework. The dataset can be represented as $((d, q), y_{(d,q)})$ composed of query-document pairs (d, q) along with the relevance assessment label denoted by $y_{(d,q)}$ which signifies the relevance of the document d to the query q. Let $c(d, q)$ be the total number of query-document pairs in the training set. Let the number of documents in the training set be D; the number of queries in the training set be Q. As adopted in [189], the confidence scores obtained from the discriminant function is used to rank documents in our proposed model. Let the words in the document d be represented by $\boldsymbol{w}^d$ and the words in the query q be represented by $\boldsymbol{w}^q$. Let the set of topics used in the document d be represented as $\boldsymbol{z}^d$, and the set of topics in the query q be represented by $\boldsymbol{z}^q$. We describe the generative process of our simplest model as follows:

1. For each topic $z = 1, \ldots, L$
   (a) Draw ϕ_z from **Dirichlet** (β)
2. For each document d in the collection D
   (a) Draw the topic proportions θ^d for each document d from **Dirichlet** (α)
   (b) For each word w_i^d in the document d
      i. Draw the topic assignment z_i^d from **Multinomial** (θ^d)
      ii. Draw a word w_i^d from **Multinomial** $(\phi_{z_i^d})$

3. For each query q in the query collection Q

   (a) Draw the per-topic query proportions θ^q from **Dirichlet** (α)

   (b) For each word w_i^q in the query q

      i. Draw the topic assignment z_i^q from **Multinomial** (θ^q)

      ii. Draw a word w_i^q from **Multinomial** $(\phi_{z_i^q})$

4. For each pair of document d and query q

   (a) Draw the response variable $y_{d,q}|z_i^d, z_i^q, (d,q), \eta$ according to Equations 8.1 to 8.3.

We will also present two other models where we maintain the order of words. We will present the derivations of our most complex model i.e. the third model, and it would not be very difficult to derive the schemes for our other two models. In Figure 8.2, we present our another model where we relax the order of words in the queries. We wish to study whether word order in short documents have any effect on the empirical results.

The generative process of our second model shown in Figure 8.2 is as follows:

1. For each topic $z = 1, \ldots, L$

   (a) Draw ϕ_{zw} from **Dirichlet** (β)

2. For each document d in the collection D

   (a) Draw the per-document topic proportions θ^d from **Dirichlet** (α)

   (b) For each word w_i^d in the document d

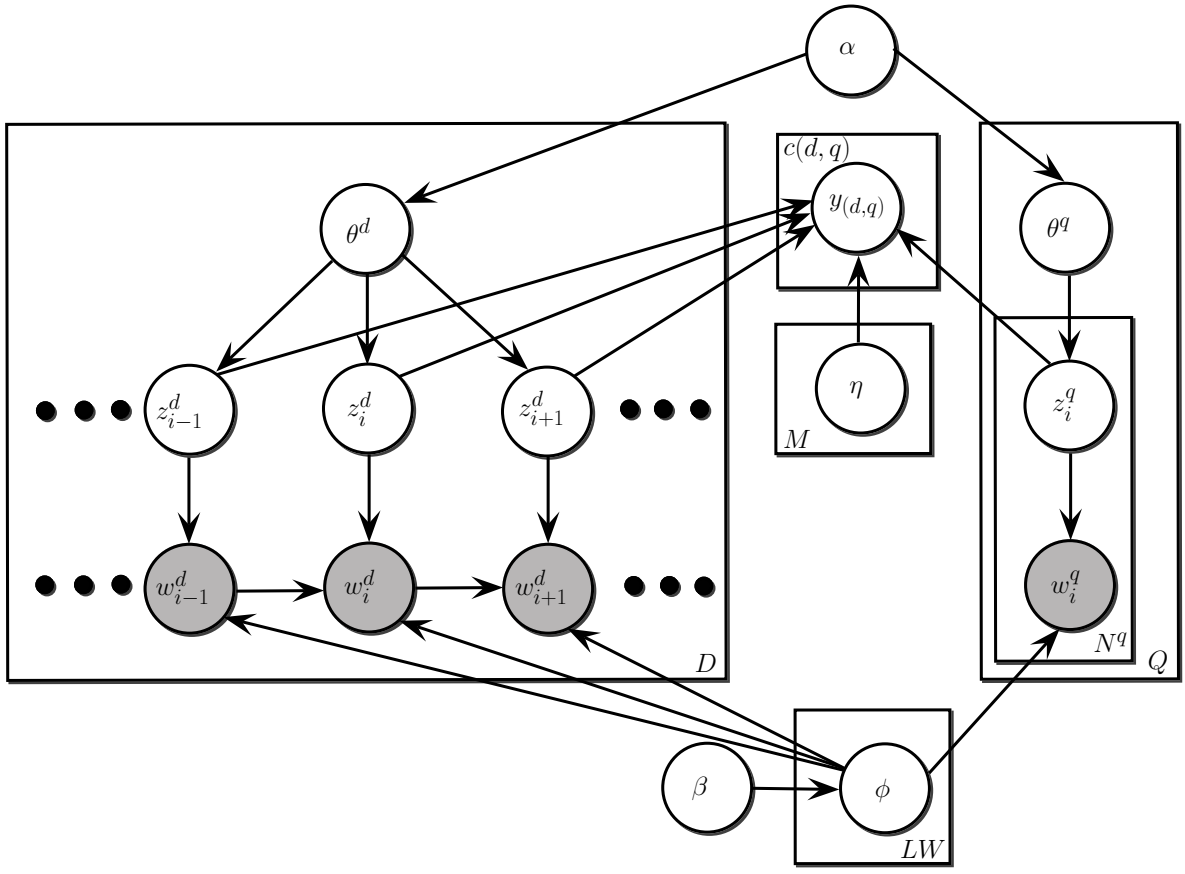


Figure 8.2: Graphical model in plate notation of our document retrieval learning model where order of words in the queries is relaxed. The model only maintains the order of words in the documents. Queries are mostly short, so it would be interesting to study whether short documents have an impact in the empirical results.

i. Draw the topic assignment z_i^d from **Multinomial** (θ^d)

ii. Draw a word w_i^d from **Multinomial** $(\phi_{z_i^d, w_{i-1}^d})$

3. For each query q in the query collection Q

   (a) Draw the per-topic query proportions θ^q from **Dirichlet** (α)

   (b) For each word w_i^q in the query q

      i. Draw the topic assignment z_i^q from **Multinomial** (θ^q)

      ii. Draw a word w_i^q from **Multinomial** $(\phi_{z_i^q})$,

4. For each pair of document d and query q

   (a) Draw the relevance label $y_{(d,q)}|(z_i^d, z_i^q, (d,q), \eta)$ according to Equations 8.1 to 8.3.

We now describe the third variant of our model. This model follows the word order in both the query and the document. The graphical model is shown in Figure 8.3.

We describe the generative process of our third model as follows:

1. For each topic $z = 1, \ldots, L$

   (a) Draw ϕ_{zw} from **Dirichlet** (β)

2. For each document d in the collection D

   (a) Draw the per-document topic proportions θ^d from **Dirichlet** (α)

   (b) For each word w_i^d in the document d

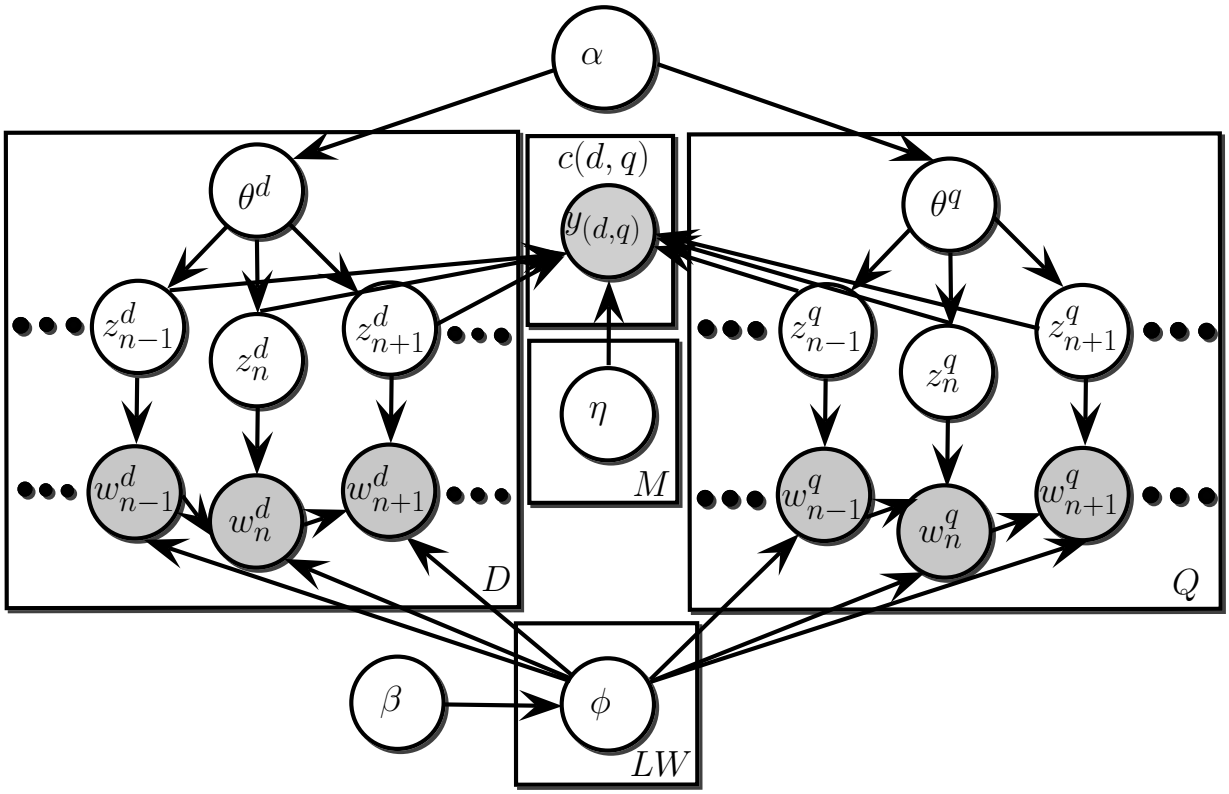


Figure 8.3: Graphical model of our document retrieval learning model with word order. The bag-of-words assumption is relaxed in both the queries and the documents.

i. Draw the topic assignment z_i^d from **Multinomial** (θ^d)

ii. Draw a word w_i^d from **Multinomial** $(\phi_{z_i^d, w_{i-1}^d})$

3. For each query q in the query collection Q

   (a) Draw the per-topic query proportions θ^q from **Dirichlet** (α)

   (b) For each word w_i^q in the query q

      i. Draw the topic assignment z_i^q from **Multinomial** (θ^q)

      ii. Draw a word w_i^q from **Multinomial** $(\phi_{z_i^q, w_{i-1}^q})$,

4. For each pair of document d and query q

   (a) Draw the relevance label $y_{(d,q)}|(z_i^d, z_i^q, (d,q), \eta)$ according to Equations 8.1 to 8.3.

The maximum-margin framework for relevance prediction follows the formulation in [123]. But in our model, each input data instance consists of a pair of document and query instead of a single document. Also in contrast to [123], the formulation of the response variable representing the relevance prediction is different.

The discriminant function of our model is designed as follows:

$$F(y, \boldsymbol{\eta}, (d,q)) = \boldsymbol{\eta}^\intercal \boldsymbol{f}(y, (d,q)) \tag{8.1}$$

where η represents the model parameters which are essentially feature weights. $\boldsymbol{f}(y, (d,q))$ is a vector of features which are designed to be useful for retrieval. We design seven features as depicted in Table 8.1. $c(w_i^d, d)$ is the number of times the word w_i^d appears in the document d. N^q is the number of words in the query q.

	Feature		Feature
1	$\sum_{w_i^q \in q \cap d} \log\left(c(w_i^q, d) + 1\right)$	4	$\sum_{w_i^q \in q \cap d} \log\left(\frac{\lvert\mathbb{D}\rvert}{c(w_i^q, d)} + 1\right)$
2	$\sum_{w_i^q \in q \cap d} \log\left(1 + \frac{c(w_i^q, d)}{\lvert d\rvert}\right)$	5	$\sum_{w_i^q \in q \cap d} \log\left(1 + \frac{c(w_i^q, d)}{\lvert d\rvert}\text{idf}(w_i^q)\right)$
3	$\sum_{w_i^q \in q \cap d} \log(\text{idf}(w_i^q))$	6	$\sum_{w_i^q \in q \cap d} \log\left(1 + \frac{c(w_i^q, d)}{\lvert d\rvert}\frac{\lvert\mathbb{D}\rvert}{c(w_i^q, \mathbb{D})}\right)$
7	Topic Similarity Feature - cosine(v^d, v^q)		

Table 8.1: Features used in our discriminant function in our document retrieval learning model.

$|.|$ denotes the length of the document or the query. idf is the inverse document frequency. The last feature, called topic similarity feature, is a similarity measure between the topics of the query and the document in the low-dimensional topic space. We first compute the average topic assignment as described in [298] separately for the document and the query. Let v^d and v^q be the topic vector for the document d and the query q respectively. This feature is formulated as a cosine similarity of v^d and v^q denoted by cosine(v^d, v^q). The first six features have also been used in [189], [274].

We can now take the expectation of the discriminant function in Equation 8.1 as follows:

$$F(y, (d, q)) = \mathbb{E}[F(y, \boldsymbol{\eta}, (d, q))] \tag{8.2}$$

The prediction rule is given in Equation 8.3.

$$\hat{y} = \underset{y}{\text{argmax}} F(y, (d, q)) \tag{8.3}$$

The following maximum-margin constraints are imposed:

$$F(y_{(d,q)}, (d, q)) - F(y, (d, q)) \geq l_{(d,q)}(y) - \xi_{(d,q)}, \forall y \in Y, \forall (d, q) \tag{8.4}$$

where $l_{(d,q)}(y)$ is a non-negative loss function. $\xi_{(d,q)}$ are non-negative slack variables

which are meant for inseparable data instances. Let $\boldsymbol{Z}^d = \{\boldsymbol{z}^d\}_{d=1}^D$ be topic assignments to all the words of the training documents; $\boldsymbol{Z}^q = \{\boldsymbol{z}^q\}_{q=1}^Q$ be topic assignments to all the words in the training queries; $\boldsymbol{\Theta}^d = \{\boldsymbol{\theta}^d\}_{d=1}^D$ be topic distributions for all training documents; $\boldsymbol{\Theta}^q = \{\boldsymbol{\theta}^q\}_{q=1}^Q$ be topic distributions for all training queries; $\boldsymbol{\Phi} = \{\boldsymbol{\phi}_{kv}\}_{v,k=1}^{W,K}$ be the word-topic distribution. C is a positive regularization constant. Let $\boldsymbol{b}^d$ and $\boldsymbol{b}^q$ denote $\{b^d_{n,n+1}\}_{n=1}^{N^d-1}$ and $\{b^q_{n,n+1}\}_{n=1}^{N^q-1}$, where $b^d_{n,n+1}$ and $b^q_{n,n+1}$ denote the words at the position n and $n+1$ in the document d and the query q respectively. Similarly $\boldsymbol{B}^d = \{\boldsymbol{b}^d\}_{d=1}^D$ and $\boldsymbol{B}^q = \{\boldsymbol{b}^q\}_{q=1}^Q$ be the word order information for the entire document collection and the query set respectively. The soft-margin framework for our model is described below:

$$
\begin{aligned}
&\underset{P(\boldsymbol{\eta},\boldsymbol{\Theta}^d,\boldsymbol{\Theta}^q,\boldsymbol{Z}^d,\boldsymbol{Z}^q,\boldsymbol{\Phi})\in\mathbb{P},\xi}{\text{minimize}} \\
&\text{KL}\ [P(\boldsymbol{\Theta}^d,\boldsymbol{\Theta}^q,\boldsymbol{Z}^d,\boldsymbol{Z}^q,\boldsymbol{\Phi})||P_0(\boldsymbol{\Theta}^d,\boldsymbol{\Theta}^q,\boldsymbol{Z}^d,\boldsymbol{Z}^q,\boldsymbol{\Phi})]- \\
&\mathbb{E}_P[\log P(\boldsymbol{B}^d,\boldsymbol{B}^q|\boldsymbol{\Theta}^d,\boldsymbol{\Theta}^q,\boldsymbol{Z}^d,\boldsymbol{Z}^q,\boldsymbol{\Phi})] + \frac{C}{c(d,q)}\sum_{(d,q)}\xi_{(d,q)} \\
&\text{subject to}\ \mathbb{E}_P[\boldsymbol{\eta}^\intercal(\boldsymbol{f}(y_{(d,q)},d,q) - \boldsymbol{f}(y,d,q,))] \geq \\
&l_{(d,q)}(y) - \xi_{(d,q)}, \xi_{(d,q)} \geq 0, \forall(d,q), \forall y
\end{aligned}
\tag{8.5}
$$

8.2.1 Posterior Inference

In order to proceed with the derivation of the collapsed Gibbs sampling, we need to define a joint distribution for words and the topics along with the regularization effects due to the maximum-margin posterior constraints. This joint distribution is written as:

$$P(\boldsymbol{Z}^d, \boldsymbol{B}^d, \boldsymbol{Z}^q, \boldsymbol{B}^q|\boldsymbol{\alpha}, \boldsymbol{\beta}) = P(\boldsymbol{B}^d|\boldsymbol{Z}^d, \boldsymbol{\beta}) \times P(\boldsymbol{B}^q|\boldsymbol{Z}^q, \boldsymbol{\beta}) \times P(\boldsymbol{Z}^d|\boldsymbol{\alpha}) \times P(\boldsymbol{Z}^q|\boldsymbol{\alpha}) \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{y=1}^{M} (\lambda_{(d,q)}^{y})^* (\boldsymbol{f}(y_{(d,q)},(d,q)) - \boldsymbol{f}(y,(d,q)))} \tag{8.6}$$

$\boldsymbol{\kappa}$ in the case of document retrieval is:

$$\boldsymbol{\kappa} = \sum_{(d,q)} \sum_{y=1}^{M} \lambda_{(d,q)}^{y} (\boldsymbol{f}(y_{(d,q)}, (d,q)) - \boldsymbol{f}(y, (d,q))) \tag{8.7}$$

After some manipulations, we can come up with the following update equation:

$$\begin{aligned} &P(\boldsymbol{Z}^d, \boldsymbol{Z}^q|\mathbf{B}^d, \mathbf{B}^q, \mathbf{Z}^d_{\neg i}, \mathbf{Z}^q_{\neg i}, \alpha, \beta) = \\ &\left(\frac{\alpha_{z_i^d} + m_{z_i^d w_i^d} - 1}{\sum_{z=1}^{K} \left(\alpha_z + m_z\right) - 1} \times \frac{\alpha_{z_i^q} + m_{z_i^q w_i^q} - 1}{\sum_{z=1}^{K} \left(\gamma_z + m_z\right) - 1} \right. \\ &\left. \times\, e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{(d,q)} \sum_{y=1}^{M} (\lambda_{(d,q)}^{y})^* (\boldsymbol{f}(y_{(d,q)},(d,q)) - \boldsymbol{f}(y_{(\hat{d},\hat{q})},(d,q)))} \right) \\ &\times \frac{\beta_{w_i^d} + m_{z_i^d w_i^d w_{i-1}^d} - 1}{\sum_{v=1}^{W} (\beta_v + m_{z_i^d w_i^d v}) - 1} \times \frac{\beta_{w_i^q} + m_{z_i^q w_i^q w_{i-1}^q} - 1}{\sum_{v=1}^{W} (\beta_v + m_{z_i^q w_i^q v}) - 1} \end{aligned} \tag{8.8}$$

where m_{zwv} be the number of times the word w is generated by the topic z when preceded by the word v and is applicable to a document and a query when super-scripted by d or q respectively. m_{zw} is the number of times a word w in the document has been sampled in the topic z, and is applicable to a document and query when super-scripted by d or q respectively.

One can argue that asymmetric priors may work better especially on short documents such as queries. Many previous works for short documents have assumed asymmetric priors in their topic models such as [277], [95], [105], [216], [124], but some have adopted symmetric priors too [110] which have shown to give good results. Our model is flexible enough to accommodate asymmetric priors, but in this paper we only test our model using symmetric priors for simplicity. In [189] the

author discussed some shortcomings in discriminative models for IR, in particular, the out-of-vocabulary words. The author has also suggested a few ways of dealing with those shortcomings. We also follow those strategies in this paper.

8.2.2 Ranking Unseen Documents

The prediction task on testing data using the prediction rule given in Equation 8.3 can be realized as follows. Let $(q^{\text{new}}, d^{\text{new}})$ be an unseen test query-document pair for which we need to predict the relevance label. The task is to compute the latent topic representations of q^{new} and d^{new} using the topic space that has been learned from the training data. These latent components for the unseen query and the document can be obtained from $\hat{\mathbf{\Phi}}$ which is the maximum aposteriori estimate of $P(\mathbf{\Phi})$ computed during the training process. Suppose there are J samples from a proposal distribution, $\hat{\mathbf{\Phi}}$ is obtained using the samples from the following equation:

$$\hat{\phi}_{zwv} \propto \frac{1}{J}\sum_{j=1}^{J}(\beta_{w_i^d} + m^{(j)}_{z_i^d w_i^d w_{i-1}^d} - 1) \times (\beta_{w_i^q} + m^{(j)}_{z_i^d w_i^d w_{i-1}^d} - 1) \tag{8.9}$$

where the counts are assigned in the j^{th} sample. The latent components for the unseen document and the query can be computed as follows.

$$\begin{aligned} &P(\boldsymbol{Z}^{d^{\text{new}}}, \boldsymbol{Z}^{q^{\text{new}}} | \mathbf{B}^{d^{\text{new}}}, \mathbf{B}^{q^{\text{new}}}, \mathbf{Z}^{d^{\text{new}}}_{\neg i}, \mathbf{Z}^{q^{\text{new}}}_{\neg i}, \alpha, \beta) \propto \\ &\hat{\phi}_{z_i^{q^{\text{new}}} w_i^{d^{\text{new}}} w_{i-1}^{d^{\text{new}}}} (\alpha_{z_i^{d^{\text{new}}}} + m_{z_i^{d^{\text{new}}}} - 1) \times \\ &\hat{\phi}_{z_i^{q^{\text{new}}} w_i^{q^{\text{new}}} w_{i-1}^{d^{\text{new}}}} (\alpha_{z_i^{q^{\text{new}}}} + m_{z_i^{q^{\text{new}}}} - 1) \end{aligned} \tag{8.10}$$

where the count for the word being sampled is excluded. We compute the similarity between the query and the document in the latent topic space. Note that $y_{(d,q)}$ can be dropped off during the prediction step. Then the expectation statistics can be

approximated as described in [123]. When the task of computing the similarity score is accomplished, it can be used in Equation 8.1 to compute the prediction score. Documents can be ranked based on this confidence score.

8.3 Retrieval Learning Experiments

8.3.1 Experimental Setup

We conduct extensive experiments on document classification using existing benchmark test collections. We also compare with many related comparative methods. In addition, we carry out qualitative analysis showing how our model generates better topical words. In all our experiments for topic models, we run the sampler for 1000 iterations. We have also removed stopwords[1] and performed stemming using Porter's stemmer[2]. Five-fold cross validation is used. In each fold, the macro-average across the classes is computed. Each model is run for five times. We take the average of the results obtained for all the runs and in all the folds.

We use three popular test collections for our experiments. We used a benchmark OHSUMED test collection (latest version[3]) from the LETOR [210] dataset. This dataset consists of 45 comprehensive features along with query-document pairs with their relevance judgments. This dataset has been used extensively in evaluating several learning-to-rank algorithms. We obtained raw documents and queries of this dataset from the web[4] in order to get the word order. The LETOR OHSUMED dataset contains the document-id along with the list of features, which will help us

[1]http://jmlr.org/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

[2]We also tested the models without performing stemming. We found that stemmed collections fared better.

[3]Minka et al. [182] and some other researchers had pointed out few shortcomings in the earlier LETOR releases.

[4]http://ir.ohsu.edu/ohsumed/

relate which set of features come from which document. In this dataset, our proposed feature i.e. the topic similarity feature is treated as one feature, in addition to the existing 45 features. It has approximately 60% query-document pairs in the training set, 20% in the validation set, and the rest in the testing set in each of the five folds. For a particular fold, the queries involved in the training, the validation, and the testing set are different. Validation set is used by the comparative learning-to-rank models for parameter tuning and determining the number of iterations. Our second collection is AQUAINT-1 used during TREC HARD[5]. Basic details about this dataset can be found in [4]. Note that we only consider document-level relevance assessments in AQUAINT-1, and leave out the passage-level judgments. The third dataset is WT2G[6], along with the standard relevance judgments and topics (401 - 450) obtained from the TREC site. Since our problem setting is similar to the pointwise learning-to-rank problem, in order to create the training, testing and validation datasets for AQUAINT-1 and WT2G, we adopted the strategies popularly used in learning-to-rank problems. We chose the same percentage of query-document pairs in the training, testing and validation set in each fold as in LETOR OHSUMED dataset. The features used for AQUAINT-1 and WT2G datasets are given in Table 8.1. Note that only the number of features differ in the datasets that we generated (WT2G and AQUAINT-1) when compared to LETOR OHSUMED. Based on our proposed model, we also investigate another variant, called `Variant 1`, which we will test empirically and show its performance. In this model we ignore the word order structure in queries, but maintain the word order structure in documents. The reason is that queries are mostly short. We use NDCG@5 and NDCG@10 as our metrics, similar to the metrics used in [36]. NDCG is well suited for our task because it is defined by an explicit position discount factor and it can leverage the judgments in terms of multiple ordered categories. In order to determine the number of topics K, the parameter C, and the loss constant function $l_{(d,q)}(y)$ in our model, we use the validation set. We first train our model on the training set, and measure

[5]http://ciir.cs.umass.edu/research/hard/guidelines2003.html
[6]http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html

NDCG@5 and NDCG@10 performance on the validation set. The number of topics and the model parameters can be determined from the validation process. We then test our model using the testing set. After the tuning process, we came up with $K = 250$ for NDCG@5, $K = 190$ for NDCG@10, $C = 100$, and $l_{(d,q)}(y) = 25$ in AQUAINT-1. Similarly, in WT2G, we obtained $K = 250$ for both NDCG@5, and $K = 170$ for NDCG@10, $C = 122$, and $l_{(d,q)}(y) = 23$. Note that we present only the average C and $l_{(d,q)}(y)$ values from five folds here for brevity in each dataset. In OHSUMED, we obtained $K = 110$ for NDCG@5 and NDCG@10 for `Our Model`, and the same number of topics for our `Variant 1`. For our `Variant 1`, we came up with $K = 230$ for NDCG@5 and $K = 190$ for NDCG@10 in AQUAINT-1, and $K = 250$ for NDCG@5 and $K = 110$ for NDCG@10 in WT2G. We have again set a weak β prior which is 0.01. We also found that varying the value of the hyperparameter does not drastically affect the results and this finding is consistent with [260]. The experimental results are averaged over five folds for all the models.

We compare the performance of our model with a range of comparative methods including popular learning-to-rank models in RankLib[7] such as `MART` [73], `RankNet` [31], `AdaRank` [274], `Coordinate Ascent` [180], `LambdaRank` [32], `LambdaMART` [270], `ListNet` [39], `Random Forests` [28] which is a popular pointwise learning-to-rank model. In addition, we used `Ranking SVM` [127][8] and `SVM`^{MAP} [286][9]. The list of first six features in Table 8.1 are also used in these comparative methods as in [189] for learning (first 45 features in case of LETOR OHSUMED). Note that the seventh feature (or 46$^{\text{th}}$ in case of LETOR OHSUMED) involves latent topic information which cannot be used in the comparative methods. In order to conduct the experiments for the comparative learning-to-rank models, we followed standard learning-to-rank experimental procedures for each comparative method. Some models have standard published parameter values, for example, for LETOR OHSUMED, the values for

[7] http://people.cs.umass.edu/∼vdang/ranklib.html
[8] http://olivier.chapelle.cc/primal/ranksvm.m
[9] http://projects.yisongyue.com/svmmap/

`Ranking SVM`[10] and `SVM`^{MAP}[11] are online.

Note that we do not choose any unsupervised topic model for comparison primarily because they cannot make use of relevance judgment information during the training process. Thus they are always at disadvantages when compared with learning-to-rank methods and our model, which explicitly uses the information of relevance labels during the training process. Also, supervised topic models such as `sLDA` cannot be directly used for comparison as one needs to make significant changes to this model to handle the document retrieval learning problem. In addition, learning-to-rank models have already shown state-of-the-art results in this task, and thus they can be regarded as strong comparative methods. Our model does not directly use word proximity features in the learning setup [173]. What our model does is to use word order for finding the best model to fit the data as it has been shown in the literature that topic models with word order improve model selection [117], [134]. Such proximity features have indeed helped improve learning-to-rank performance, but in this work our objective is to present the robustness of our model.

8.3.2 Quantitative Results

We present results obtained from all the three test collections in Table 8.2. From the results, we can see that our model outperforms all the comparative methods. The improvements that we obtain are statistically significant according to Wilcoxon signed rank test (with 95% confidence) against each of the comparative methods in Table 8.2 on all the datasets. Our results show that the latent topic information generated by our model which is then used to compute query-document similarity plays a significant role. Word order too plays a role where we are able to detect better topics than unigram models.

[10] http://research.microsoft.com/en-us/um/beijing/projects/letor/baselines/ranksvm-primal.html

[11] http://www.yisongyue.com/results/svmmap_letor3/details.html

	OHSUMED		AQUAINT-1		WT2G	
Models	NDCG@5	NDCG@10	NDCG@5	NDCG@10	NDCG@5	NDCG@10
`Our Model`	0.483	0.461	0.454	0.460	0.511	0.511
`Variant 1`	0.474	0.460	0.450	0.452	0.448	0.473
`MART`	0.420	0.403	0.355	0.380	0.433	0.447
`RankNet`	0.395	0.384	0.401	0.408	0.313	0.370
`RankBoost`	0.424	0.436	0.423	0.433	0.395	0.401
`AdaRank`	0.469	0.445	0.406	0.418	0.409	0.431
`Coordinate Ascent`	0.472	0.455	0.422	0.438	0.433	0.450
`LambdaRank`	0.354	0.278	0.218	0.287	0.211	0.217
`ListNet`	0.443	0.441	0.366	0.376	0.245	0.226
`Random Forests`	0.380	0.411	0.415	0.421	0.422	0.432
`Ranking SVM`	0.461	0.454	0.365	0.387	0.367	0.372
`LambdaMART`	0.447	0.437	0.352	0.367	0.326	0.367
$\texttt{SVM}^{MAP}$	0.453	0.432	0.389	0.401	0.406	0.415

Table 8.2: The performance of our model in comparison to other learning-to-rank models.

One interesting facet to consider is to study the effect of the number of topics in the document retrieval learning experiment for our models. In Table 8.3, we vary the number of topics from 50 to 290 in steps of 20 and present the results therein. In the OHSUMED dataset we can see that as we increase the number of topics, the results improve until certain number of topics and begin to deteriorate again as we keep on increasing the number of topics. This gives us an insight about the dependence between the number of topics and the retrieval learning results for our models. We can also see that our `Variant 1` model does not perform very well for most of the cases suggesting that word order plays a dominant role. Similar trend is also observed in AQUAINT-1 and WT2G collections where we see that the results improve until certain point, and then begin to deteriorate again. Our `Variant 1` model still remains less effective against our proposed model.

It is quite interesting to see that our model outperforms some of the powerful learning-to-rank models. Our model can perform consistently well with more (in LETOR OHSUMED) and less number of features (in WT2G and AQUAINT-1).

		OHSUMED		AQUAINT-1		WT2G	
Topics	Metric	Our Model	Variant 1	Our Model	Variant 1	Our Model	Variant 1
50	NDCG@5	0.345	0.344	0.382	0.381	0.395	0.302
	NDCG@10	0.356	0.345	0.386	0.382	0.401	0.315
70	NDCG@5	0.401	0.400	0.389	0.381	0.433	0.431
	NDCG@10	0.367	0.367	0.395	0.396	0.437	0.432
90	NDCG@5	0.455	0.446	0.401	0.411	0.415	0.408
	NDCG@10	0.406	0.410	0.437	0.439	0.452	0.446
110	NDCG@5	0.483	0.474	0.450	0.450	0.481	0.472
	NDCG@10	0.461	0.460	0.459	0.455	0.483	0.473
130	NDCG@5	0.481	0.481	0.451	0.451	0.495	0.481
	NDCG@10	0.461	0.463	0.459	0.459	0.499	0.485
150	NDCG@5	0.482	0.486	0.451	0.449	0.501	0.486
	NDCG@10	0.466	0.461	0.459	0.458	0.504	0.485
170	NDCG@5	0.473	0.463	0.451	0.449	0.523	0.492
	NDCG@10	0.461	0.456	0.460	0.459	0.511	0.495
190	NDCG@5	0.465	0.459	0.452	0.448	0.523	0.491
	NDCG@10	0.445	0.442	0.460	0.452	0.512	0.495
210	NDCG@5	0.462	0.456	0.452	0.447	0.525	0.493
	NDCG@10	0.443	0.441	0.460	0.453	0.511	0.496
230	NDCG@5	0.441	0.420	0.455	0.450	0.524	0.485
	NDCG@10	0.430	0.413	0.460	0.452	0.520	0.486
250	NDCG@5	0.421	0.411	0.454	0.450	0.511	0.448
	NDCG@10	0.398	0.395	0.461	0.455	0.514	0.455
270	NDCG@5	0.423	0.401	0.452	0.448	0.492	0.432
	NDCG@10	0.398	0.365	0.460	0.455	0.499	0.441
290	NDCG@5	0.444	0.399	0.452	0.447	0.484	0.432
	NDCG@10	0.356	0.356	0.460	0.451	0.483	0.442

Table 8.3: Results obtained from our models when the number of topics is varied.

This shows that the generalization ability of our proposed model is very robust. The results suggest that incorporating topic similarity helps improve document retrieval performance. One reason why topic models help improve document retrieval performance as we compare the similarity between the document and the query based on latent factors rather than just the words [266], [236]. Hence, this feature which our model computes is extremely important for document retrieval task.

8.3.3 Qualitative Analysis

Our qualitative results are very strong as well (presentation of words follow the same procedure as described in Section 7.3.3). We can see from Table 8.4, our model has generated words which appear more meaningful that the rest of the models. We have

BTM	LDACOL	TNG	PDLDA	NTSeg	Our Model
foreign beggars	today	news corp	foreign minister stevo	today	news viewership
bt anton	hebron	www	fundamental prerequisites	atlanta	foreign minister
hk salem	bosnian	web	jewish state restarts	hk salem	president nasser
fundamental prerequisites	foreign beggars	news event	reported exceptionally	york times news service	general news
great stash	atlanta	york steaks	york times news service	bosnia	resistance occurred

Table 8.4: Top five probable words from a topic from AQUAINT-1 collection.

only considered words from documents in order to present results in this table. Our model uses query-document relevance label for generating words.

8.4 Closing Remarks

In this chapter, we have presented our supervised topic models to document retrieval learning task. This model takes as input the query-document pairs. Relevance assessments given manually by annotators are the response variables. The model computes the query-document topic similarity in the low-dimensional latent topic space, which is then used as one of the features in the discriminative learning-to-rank framework. We saw through experimental analysis that our model outperformed many popular learning-to-rank models. We have also seen that by relaxing the order of words in the queries degrades the performance of the model. Through qualitative analysis we presented how our model generates better topical words than the comparative methods.

Chapter Nine

Readability Prediction and Ranking

Chapter Summary

In this chapter, we propose a novel framework for determining the conceptual difficulty of a domain-specific text document without using any external lexicon. Conceptual difficulty relates to finding the reading difficulty of domain-specific documents. Previous approaches to tackling domain-specific readability problem have heavily relied upon an external lexicon, which limits the scalability to other domains. Our model can be readily applied in domain-specific vertical search engines to re-rank documents according to their conceptual difficulty. We develop an unsupervised and principled approach for computing a term's conceptual difficulty in the latent space.

9.1 The Case for the Terrain Models in LSI

Definition 6. *Readability: It relates to the nature or quality of a text document that makes it easy or difficult to understand and read.*

Definition 7. *Domain: It is defined as a specific subject area in a particular field.*

Definition 8. *Domain-expert: A person who has a thorough background of a domain.*

Definition 9. *Domain-specific Readability: It relates to nature or quality of a domain-specific text document that makes it easy or difficult to understand and read.*

Definition 10. *Domain-specific concept or word: It is a word or a phrase which has a specific meaning in a domain. For example, "Random access memory" is a domain-specific concept. It is also used as a technical term at some places in this thesis.*

Definition 11. *Conceptual Difficulty: It relates to the difficulty in understanding a domain-specific word.*

Technical terms play a deep-seated role in determining the expertise level of a document. Consider two examples which exemplify the above mentioned observation.

Example 1

1. A chemical bond is an interaction between atoms or molecules and allows the formation of polyatomic chemical compounds.

Example 2

2. Chemical bond refers to the forces holding atoms together to form molecules and solids .

We can see that the first text comprises of domain-specific terms which are meant for experts in the field, whereas the second text explains the idea of a "chemical bond" in simple terms. We have highlighted some portions in both the examples. In the first example, we note that most of the highlighted words may appear difficult to a beginner as they are domain-specific terms. In order to grasp what the definition of a "Chemical bond" means in the first example, one needs to have some background knowledge in Chemistry. In contrast, the second definition is for the general readers who have little or no background knowledge in Chemistry. We have also highlighted some words in the second example which contain very simple description of the idea of a "Chemical bond". Our goal is to design a computational model which can automatically learn from the data the reading difficulty, and predict the reading difficulty of new documents from the learned model under an unsupervised setting.

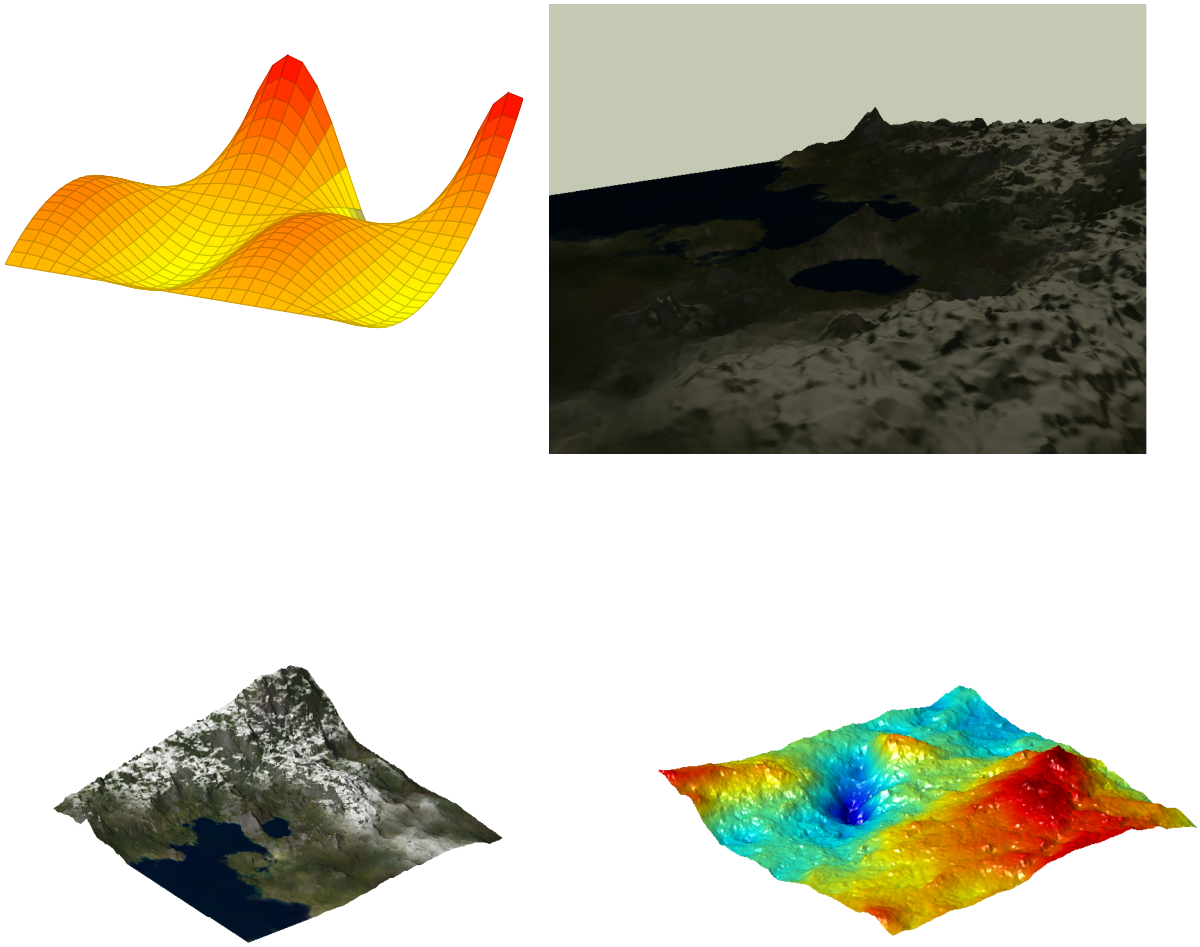


Figure 9.1: Figures depicting different terrain types. A terrain appears to be bumpy with plenty of ups and down. A person walking on such a terrain landscape has to face difficulty in going from one point to another owing to a large number of bumps in the path. Similar is the idea of our readability model where a reader faces more difficulty if the text that the reader has to read in a document consists of difficult domain-specific concepts.

Latent semantic models, such as, NMF [152] and LSI [58], [13] have not been explored to tackle the problem of readability. The first latent semantic model introduced in the field of IR is the LSI. In fact, the LSI was the first model to generate concepts which later came to be popularly known as topics. Models such

as Probabilistic Latent Semantic Indexing (pLSI) [102] and pLSA models [101] and LDA have been inspired from this model. The LSI model can better be called as a concept model rather than a topic model. We present a model in this thesis where we apply the LSI model for retrieval of domain-specific documents based on readability with the consideration of word order. Works which have looked into the problem of domain-specific readability [279], [293], [292] have remained constrained to certain domains only, for instance, the Medical domain because of the required reliance on some knowledge bases to find domain-specific terms in a document. To circumvent this limitation, we discuss a novel unsupervised framework for computing domain-specific document readability. The main factor that makes our proposal superior compared with the existing domain-specific readability methods is that our method does not require an external ontology or lexicon of domain-specific terms. We have also presented different terrain models [118], [119], [120], [121] in LSI which predict the technical difficulty of documents in domain-specific IR. These models are based on conceptual hops between the unigrams. We have also presented an n-gram extension to the document terrain modeling in [122]. The aggregated cost obtained in the end of the document completion corresponds to the readability cost of the document.

We have developed novel terrain models in the LSI space which maintains the document's semantic structure in order to compute the reading difficulty of a document. We depict terrains in Figure 9.1. The idea is to convert the high dimensional vector space of terms and documents to a low-dimensional concept space which brings out better word-document and word-word correlations. Then convert the original document space into a series of cliffs just like a terrain where a high distance between the low-dimensional term vectors in sequence in the documents signifies the high energy that one has to spend in order to cover a conceptual leap, and the difficulty of a term as the cliff which a reader has to climb. The higher the term score, higher will be the cliff, and hence more difficult a concept.

9.2 The Terrain Model in the Concept Space

9.2.1 Sequential Term Transition Model (STTM)

Overview

Our proposed framework which we term as Sequential Term Transition Model (STTM) considers two components for determining the accumulated conceptual difficulty of a text document. These two components are technical term difficulty and sequential segment cohesion. Reading difficulty of a document is directly proportional to individual term's difficulty. The more cohesive the terms are, the more technically simple a document will be [279]. We group multiple terms in sequence into variable length segments and measure similarity between the sequences of segments in a document.

The idea behind our model can be visualized in this way. Consider a person trying to pass a terrain. If there are high cliffs and ridges, then the amount of energy that the person has to expend will be more. If the region which a person is passing is not full of high cliffs and ridges, then the energy expended will be less. So more the energy spent, more is the difficulty to cross the terrain and vice-versa.

The Latent Space

We make use of LSI to derive latent information that plays a major role in our framework. One essential component in LSI is SVD. Consider a domain, the input to LSI is a $W \times D$ term-document matrix, $\mathbf{W}$, where W is the number of terms in the vocabulary, and D is the number of documents in the collection. The term-document matrix can be constructed by considering the product of Term-Frequency (TF) and Inverse Document Frequency (IDF). SVD factorizes $\mathbf{W}$ into three matrices as shown

in Equation 9.1.

$$\mathbf{W} \approx \hat{\mathbf{W}} = \mathbf{USV}^T \tag{9.1}$$

where $\mathbf{U}$ is a $W \times f$ matrix of left singular vectors, $\mathbf{S}$ is a $f \times f$ diagonal matrix of singular values, $\mathbf{V}$ is a $D \times f$ matrix of right singular vectors, where $f << \min(W, D)$ is the number of factors, and $\mathbf{V}^T$ denotes matrix transposition of $\mathbf{V}$. One can also refer to the Background section (Chapter 3) for more details.

Traditional Vector Space Model (VSM) [225] cannot find new structural relationships between terms and their documents in the collection [11]. By considering SVD, one of our aims is to reduce the dimension of the space, and thus reduce the effect of noise. Moreover, considering this scheme will help bring close some terms which are coherent to the document. For example, if a document describes about Astronomy, terms such as "star" will come close to the document in the latent space [118], [120], [119], [121], [122]. Then we can compute the domain-specific importance or its difficulty in the document which is not possible to measure using readability methods because of their reliance in determining difficulty of terms using surface level features.

Technical Term Difficulty

Every term in a domain-specific document is characterized by certain difficulty in a domain. Some terms such as technical terms of a domain are shared less among documents as they are not commonly used; compared to the common terms such as "because", "composed" etc, which are common/general terms used in everyday language. The notion of technical term difficulty is similar to the notion of document scope in [279] where difficulty of a concept is measured based on the depth of a

concept in the ontology tree. In contrast, we measure scope of a domain-specific term without an ontology.

We formulate the notion of computing a term's difficulty as a term embedding problem which embeds a term vector by a weighted linear combination of document vectors in the latent space. The low-dimensional representation of term and document vectors obtained via SVD is not normalized. Normalization ensures numerical stability of our model and closeness is completely measured by angles between the vectors and the effect of diverse magnitude is discarded.

Recall the SVD factorization as described in Equation 9.1. Suppose that $\mathbf{U}$ and $\mathbf{S}$ are matrices in SVD computation as expressed in Equation 9.1. Let $\mathbf{R}$ be a matrix with dimension $W \times f$ and $\mathbf{R}$ is computed by matrix multiplication of $\mathbf{U}$ and $\mathbf{S}$ as depicted in Equation 9.2.

$$\mathbf{R} = \mathbf{U} \times \mathbf{S} \tag{9.2}$$

Let $\vec{r}_x$ denote the term vector at row x in matrix $\mathbf{R}$. The dimension of $\vec{r}_x$ is $1 \times f$. Equivalently, $\mathbf{R}$ can be expressed as in Equation 9.3.

$$\mathbf{R} = \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \cdots \\ \vec{r}_x \\ \cdots \\ \vec{r}_T \end{bmatrix} \tag{9.3}$$

We normalize $\vec{r}_x$ as follows:

$$\hat{\vec{r}}_x = \frac{\vec{r}_x}{||\vec{r}_x||} \tag{9.4}$$

Let $\mathbf{L}$ be a matrix of dimension $f \times D$ and $\mathbf{L}$ is computed by a matrix multipli-

cation of $\mathbf{S}$ and $\mathbf{V}^T$ as depicted in Equation 9.5

$$\mathbf{L} = \mathbf{S} \times \mathbf{V}^T \tag{9.5}$$

Let $\vec{l_j}$ denote a document vector at column j in $\mathbf{L}$. The dimension of $\vec{l_j}$ is $1 \times f$. Equivalently, $\mathbf{L}$ can be expressed as depicted in Equation 9.6.

$$\mathbf{L} = \begin{bmatrix} \vec{l_1} \\ \vec{l_2} \\ \cdots \\ \vec{l_j} \\ \cdots \\ \vec{l_D} \end{bmatrix}^T \tag{9.6}$$

We normalize each document vector $\vec{l_j}$ as depicted in Equation 9.7.

$$\vec{\hat{l}_j} = \frac{\vec{l_j}}{||\vec{l_j}||} \tag{9.7}$$

In our approach, for each term in the vocabulary, we attempt to compute a scale factor associated with each document in which the term exists. Consider a term x from the vocabulary. Let the index set of documents that contain term x be denoted as $\{q_1, q_2, \cdots, q_N\}$ where N is the total number of the documents that contain the term x. We construct a matrix $\hat{\mathbf{L}}_x$. Each row in $\hat{\mathbf{L}}_x$ corresponds to document vector $\vec{\hat{l}}_{q_i}$. The dimension of $\hat{\mathbf{L}}_x$ is $N \times f$. As a result, $\hat{\mathbf{L}}_x$ can be expressed as depicted in Equation 9.8

$$\hat{\mathbf{L}}_x = \begin{bmatrix} \vec{l}_{q_1} \\ \vec{l}_{q_2} \\ \cdots \\ \vec{l}_{q_N} \end{bmatrix} \tag{9.8}$$

The term linear embedding problem can be formulated as minimizing the distance expressed in Equation 9.9.

$$\begin{aligned} \underset{[\gamma_n^x]}{\text{minimize}} \quad & ||\vec{\hat{r}}_x - [\gamma_n^x]^T \hat{\mathbf{L}}_x|| \\ \text{subject to} \quad & \sum_{n=1}^{N} \gamma_n^x = 1, \gamma_n^x \geq 0 \end{aligned} \tag{9.9}$$

The weights encapsulated in $[\gamma_n^x]$ by linear synthesis in Equation 9.9 can be regarded as technical contribution that the term plays in the document. The dimension of $[\gamma_n^x]$ is $N \times 1$. By adopting the optimization in Equation 9.9, we are finding a scale factor $[\gamma_n^x]$ associated with document n for term x such that the scaled vector $[\gamma_n^x]^T \hat{\mathbf{L}}_x$ is as close as possible to the term vector $\vec{\hat{r}}_x$. The linear combination coefficients of each document synthesized with the term are in $[\gamma_n^x]$. The coefficient will obtain a higher value, if the document vector is close to the term vector in the latent space. The coefficients will be low when the document is far from the term. Therefore, domain-specific terms will come close to the document vector in the latent space. The closer they are, the rarer they are in the document collection and therefore an average reader will find the term difficult to comprehend.

We conduct optimization expressed in Equation 9.9 for each term in the vocabulary. Consider document j from the entire collection. Let N^d be the total number of terms in document j. Every term t_i in j will have a conceptual difficulty value denoted as $\gamma_j^{t_i}$. Then the difficulty score, χ_j of the document j can be formulated as:

$$\chi_j = \frac{\sum_{i=1}^{N^d} \gamma_j^{t_i}}{N^d} \tag{9.10}$$

Sequential Segment Cohesion

As [91] pointed out that document displays varying degree of cohesion. The beginning of text will not be cohesive with the later sections of the same text. The main hurdle in technical comprehensibility comes when a reader has to relate different technical storylines occurring in sequence both of which deal with different thematic interpretations in the same document. Here a *segment* is referred to multiple terms in sequence which belong to the same cluster in the LSI latent space. This notion is different from text segmentation approaches where the prime focus is to measure change in the thematic ideas or topics in text [10], [117]. Our approach mainly considers change in the concept cluster membership in latent concept space and the segment lengths may be smaller in length compared with traditional text segmentation approaches.

Input: Collection of text documents, cluster information of terms.
Output: Cohesion score of a document.

$\zeta_j \leftarrow 0$;
$S_j \leftarrow 1$;
$t_i \leftarrow$ READ a unigram from document;
$\pi_i \leftarrow$ GetClusterMembership(t_i);
$\Delta_i \leftarrow$ GetClusterCentroid(π_i);
while *not at the end of this document* **do**
  $t_{i+1} \leftarrow$ READ a unigram from document;
  $\pi_{i+1} \leftarrow$ GetClusterMembership(t_{i+1});
  $\Delta_{i+1} \leftarrow$ GetClusterCentroid(π_{i+1});
  if $(\pi_i \neq \pi_{i+1})$ **then**
    $\alpha \leftarrow \nu(\Delta_i, \Delta_{i+1})$;
    $\zeta_j \leftarrow \zeta_j + \alpha$;
    $\Delta_i \leftarrow \Delta_{i+1}$;
    S_j=$S_j + 1$;
    $\pi_i = \pi_{i+1}$;
  else
    Go back to the beginning of the loop;
  end
end
return$\left(\frac{\zeta_j}{S_j} \times \tau\right)$;

Algorithm 3: Cohesion based on segmentation.

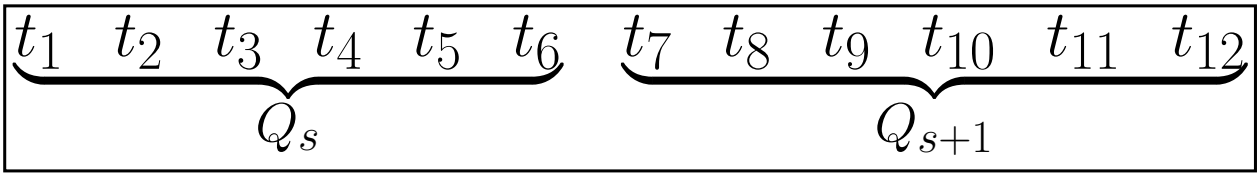


Figure 9.2: One particular term sequence $(t_1, t_2, \cdots, t_{10})$ with two segments (Q_s, Q_{s+1}) in sequence.

Generally the latent space obtained via SVD does not directly provide a reasonable cluster membership of every term in space [275]. A clustering algorithm is needed. In [11], k-means is applied followed by bottom up clustering to determine the cluster membership of terms in the latent space. We adopt similar clustering technique because k-means is well suited for handling large datasets as ours [114]. We cluster low-dimensional term vectors in the latent space. The input to the clustering algorithm are the normalized low-dimensional term vectors $\vec{\hat{r}}_x$ as depicted in Equation 9.4.

A segment is a sequence of terms in the document which belong to the same conceptual cluster in the latent space. We show one such example in Figure 9.2, where Q_s represents a segment. Our model for finding cohesion is to traverse the sequence of terms in order in the latent concept space. We call this process "conceptual transitions" in latent space. We keep moving forward in sequence until a change in cluster membership of a term occurs. Let $\nu(\vec{\Delta}_s, \vec{\Delta}_{s+1})$, denote cosine similarity between the centroids of two clusters to which the segments belong, where $\vec{\Delta}_s$ represents the centroid of the cluster in which a segment Q_s exists, and $\vec{\Delta}_{s+1}$ represents the centroid of the cluster of the next segment. Let S_j be the total number of segments in document j and τ be the average number of terms of all segments in the document. Let ζ_j denote the overall cohesion score of document j. The cohesion score of document j is formulated as:

$$\zeta_j = \frac{\sum_{s=1}^{S_j - 1} \nu(\vec{\Delta}_s, \vec{\Delta}_{s+1})}{S_j} \tau \tag{9.11}$$

If the document is cohesive, then majority of the terms in document will belong to a single segment and τ will have a high value. If terms are not semantically associated with each other in the discourse, number of segments S_j will be high in the document. As a result, overall cohesion will be lowered indicating that document is conceptually difficult. Hence, at each forward traversal in the document, a reader will experience certain amount of conceptual leaps.

We show the steps for computing cohesion as a pseudo-code in Algorithm 3. We traverse the sequence of terms in a text document and at each forward movement, ascertain the cluster membership of term t_i in sequence (procedure **GetClusterMembership()**). If the sequences of terms come from the same cluster, this indicates that terms in sequence are cohesive. We keep on traversing forward until a change in the term's cluster membership occurs which indicates weakness in cohesion among terms in sequence. We keep track of the number of segments in S_j. We measure segment cohesion by computing the cosine similarity (procedure **CosineSimilarity()**) between the two centroids (procedure **GetClusterCentroid()**) of the clusters to which the two segments belong. In the end, this will result in the document being segmented into several different segments each of which incorporates one cohesive group of terms and cohesion score of the document is aggregated.

9.2.2 Document Conceptual Difficulty Score

Our approach determines the relative "conceptual difficulty" of a document when hopping/traversing through text sequentially, where difficulty of documents is measured in the latent space that represent a deviation from the common terms and cohesion between the segments. The overall conceptual difficulty of a document will be directly proportional to individual difficulties of each term in the document and inversely proportional to cohesion score. The more the cohesion among the units of text, the lesser will be the conceptual difficulty in comprehending a technical dis-

course [140]. Therefore, conceptual difficulty, Φ_j of a document can be formulated as:

$$\Phi_j = \beta\chi_j + (1-\beta)\frac{1}{\zeta_j + 1} \tag{9.12}$$

where β $(0 \leq \beta \leq 1)$ is the parameter controlling the relative contribution between term difficulty and cohesion. We have added 1 in the denominator of cohesion score to handle the case when the centroids are orthogonal to each other. Φ_j gives an indication about the conceptual difficulty of document j. This score will be used to re-rank the search results obtained from a similarity based IR system.

9.2.3 Experiments and Results

Data Preparation

Existing standard IR test collections such as those used in Text Retrieval Conference (TREC) and Cross-Language Evaluation Forum (CLEF) cannot fulfill our purpose of evaluation as we need conceptual difficulty judgment on each document. Hence we collected a large test collection of web pages of our own as done by the topical search engines. To ascertain the full operational characteristics of our model, we chose Psychology domain. We crawled a large number of web pages from various resources. Enlisting every crawled source would be too long but we name a few popular sites from where we crawled web pages: 1) Wikipedia, 2) Psychology.com, 3) Simple English Wikipedia, and some more related web sites. We crawled 167,400 web pages with 154,512 unique terms in the vocabulary. No term stemming was performed. We prepared two sets of documents, one with stopwords[1] kept and another with stopwords removed. Removing stopwords breaks the natural semantic structure of the document, but this will capture conceptual leaps between the sequences of content words.

[1] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stoplist/english.stop

To collect queries that an average user is likely to use for searching information about a domain, we followed the INEX[2] topic development guidelines. However, our topic creators were not domain experts. In all, we had 110 topics. Some sample information needs are: 1) depression, 2) fear of flying 3) intimacy.

Experimental Setup

We refer our model with stopwords kept as STTM(Stop) and with stopwords removed as STTM(No Stop). One of our aims was to test the role of stopwords in determining the conceptual difficulty of documents. We compared with other state-of-the-art approaches in terms of conceptual difficulty prediction and ranking. We used Zettair[3] to conduct retrieval and obtained a ranked list using Okapi BM25 [221] ranking function. We then selected top ten documents for evaluation purpose. The reason for selecting these documents for evaluation is that we observed that these documents from Zettair system were all relevant to the query and the list contained a mix of documents with different conceptual difficulty. These documents were then re-ranked automatically from conceptually simple to difficult using our proposed models as well as some existing models for comparison. Similar kind of experimental setup and document re-ranking scheme have been adopted in [279], [188]. The reason for re-ranking from conceptually simple to advanced in our experiments is as follows. According to the studies undertaken relating to the behavior of novices and expert searchers, it has been found that an increasing number of users are searching for information in unfamiliar domains [14]. Hence, most of them will probably look for introductory level documents. A study has also found that domain experts employ complex search strategies such as usage of jargon, complex phrases to successfully retrieve documents matching their expertise level [269]. Therefore, ranking from conceptually simple to advanced fits most of the users. As stated previously in [279], [188], the authors also ranked documents from introductory to advanced when they

[2]http://www.inex.otago.ac.nz/tracks/adhoc/gtd.asp
[3]http://www.seg.rmit.edu.au/zettair/index.html

tested their model on users possessing average level of knowledge about healthcare. In [278], the authors re-ranked documents based on decreasing specificity.

We have set the value of $\beta = 0.5$ in our experiments which means that equal weights are given to both components. The value of k in k-means was 150. We have set $f = 200$ (defined in Section 9.2.1) because in general low number of factors are ideal for effective results [65]. We used SeDuMi with YALMIP [169] to conduct optimization in Equation 9.9. Our main model is STTM(Stop) because our objective is to test our model on the entire document structure without removing any of the features.

The existing unsupervised methods used as comparative methods include: 1) Okapi BM25 described in [221], 2) Dirichlet smoothed, query likelihood language model [288] (denoted as LM) with default parameter as in Zettair), and 3) Cosine similarity based retrieval [225]. In addition, we also compared with widely used unsupervised readability scores, namely, 1) ARI, 2) Coleman-Liau (C-L) (denoted as C-L in the tables), 3) Flesch Reading Ease formula, 4) Fog, 5) LIX, and 6) SMOG. More details about readability methods can be found in [64]. For each readability formula it computes a readability score for every document. Then the documents are re-ranked in descending order of the readability score. We also compare our model against one of our previously proposed methods Conceptual Hop Model (CHM) described in [118]. Our model works by considering only the semantic content of text. Readability methods contain both semantic and syntactic components. Therefore, we only chose the semantic component of readability methods.

It is important to note that readability methods and traditional ranking methods form the most suitable comparative methods because they are completely unsupervised. Domain-specific readability methods such as [279], [293] use an extra lexicon of technical terms.

Annotation Guidelines	
The relative technical difficulty of the document that you are currently reading is - ?	
4	Very low
3	Reasonably low
2	Borderline
1	Reasonably high
0	Very high

Table 9.1: Conceptual difficulty judgment guidelines given to the human judges.

Evaluation Metric

To obtain a ground truth of conceptual difficulty of documents for evaluation purpose, two human annotators who were undergraduate students having varied background were invited. They had basic knowledge about Psychology. The annotators were fluent in reading English passages. They gave annotations following guidelines given in Table 9.1. They were also asked to read the articles sequentially without skipping any term in the document. In the beginning we acquainted them with the main aim of the study and also showed them some sample documents from our test collection so that they could get an idea about the relative difficulty levels of documents in the collection. The standard deviation of judgments among the annotators was 1.23.

We evaluate our method using NDCG. NDCG is widely used for IR ranking effectiveness measurement. NDCG is well suited for our task because it is defined by an explicit position discount factor and it can leverage the judgments in terms of multiple ordered categories. NDCG@i scores will directly correlate with the difficulty annotation of documents given by humans. Such scores can measure the quality of difficulty ranking of documents based on the difficulty judgments provided by humans with levels shown in Table 9.1. If NDCG is high, it means that the ranking function correlates better with the human judgments. The formula for NDCG is:

$$N(q_s) = \frac{1}{J_n} \sum_{i=1}^{n} \frac{2^{\mathrm{r(i)}} - 1}{\log(1+i)} \tag{9.13}$$

Method	NDCG@3	NDCG@5	NDCG@7	NDCG@10
Okapi BM25	0.429	0.462	0.500	0.526
LM	0.433	0.465	0.502	0.529
Cosine	0.542	0.581	0.599	0.654
STTM(Stop)	0.579	0.600	0.640	0.670
STTM(No Stop)	0.576	0.599	0.641	0.669

Table 9.2: Ranking performance of popular ranking models at different retrieval points. STTM has obtained a statistically significant result according to paired t-test ($p < 0.05$) against all models. STTM(Stop) is our model with stopwords kept and STTM(No Stop) is our model with stopwords removed. We have set $\beta = 0.5$ defined in Equation 9.12 so that equal weights come from both components of our model.

where r(i) is the rank label of the i^{th} document in the ranked list, n is the length of the ranked list, J_n is the normalization constant such that a perfect list gets a score of 1.

9.2.4 Results Discussion

We present the main result in Tables 9.2 and 9.3. Our model has significantly outperformed (using paired t-test $p < 0.05$) traditional ranking functions in Table 9.2 and it matches our general intuition that the traditional ranking functions are not suitable for handling ranking of documents based on conceptual difficulty. One notable observation is the role of stopwords in our results. One can notice that STTM(Stop) has relatively performed better than STTM(No Stop) in our experiments. Importance of stopwords has also been studied in [145] where the authors found out that stopwords have played an important role in their Familiarity Classifier (FAMCLASS) classifier.

In Table 9.3 we compare our model against widely used readability formulae. Our model has also performed significantly better than any other comparative method (using paired t-test ($p < 0.05$)). This points to the fact that readability formulae fail to differentiate terms based on contextual usage and their difficulties. In Table 9.4, we present query-wise performance of our model compared with the comparative

Method	NDCG@3	NDCG@5	NDCG@7	NDCG@10
ARI	0.515	0.548	0.582	0.618
C-L	0.525	0.553	0.584	0.612
Flesch	0.449	0.490	0.537	0.579
Fog	0.513	0.547	0.577	0.612
LIX	0.516	0.550	0.584	0.619
SMOG	0.517	0.550	0.579	0.616
CHM	0.465	0.456	0.473	0.482
STTM(Stop)	0.579	0.600	0.640	0.670
STTM(No Stop)	0.576	0.599	0.641	0.669

Table 9.3: Ranking performance of our models against popular readability models at different retrieval points. STTM has obtained a statistically significant result according to paired t-test ($p < 0.05$) against all models.

Method Name	Queries Improved		Average Improvement	
	STTM(Stop)	STTM(No Stop)	STTM(Stop)	STTM(No Stop)
Okapi BM25	60	56	34.56%	30.45%
LM	59	53	32.71%	27.66%
Cosine	43	38	19.93%	14.91%
ARI	48	39	12.34%	10.12%
C-L	56	48	16.23%	12.43%
Flesch	58	40	15.65%	11.33%
Fog	58	50	16.44%	8.34%
LIX	51	43	13.98%	7.55%
SMOG	40	38	13%	9.46%
CHM	71	68	33%	23.54%

Table 9.4: Query-wise performance of our model compared with the comparative models.

methods. It can be seen that in most of the cases our model outperforms the comparative methods by a high margin. CHM did not perform very well due to a weak non-linear model.

We experimented STTM by varying $0 \leq \beta \leq 1$ in Equation 9.12. We show results in Figure 9.3. We have obtained statistically significant results using paired t-test ($p < 0.05$) across all values for β against all methods. What can be observed from the two ends of the abscissa in Figure 9.3 is that a β close to 0 attains greater NDCG@10. The contribution from difficulty is more uniform across all documents than from cohesion. In other words, the usage of the terminologies is at the same level.

Through our study we have found that traditional ranking functions are not designed to handle ranking by difficulty of documents. We have also found that the readability formulae are not directly applicable to the problem of determining the conceptual difficulty of documents. What makes our model superior when compared with other models is that we are able to effectively capture term difficulties of the domain-specific terms based on their contextual information. It means that in one technical discourse, if a term is used as a general term, its difficulty will be low. However the same term whose semantic fabric coherently matches with the technical storyline of the document will have a high conceptual difficulty score. Our model also captures conceptual leaps during sequential term traversal in the document.

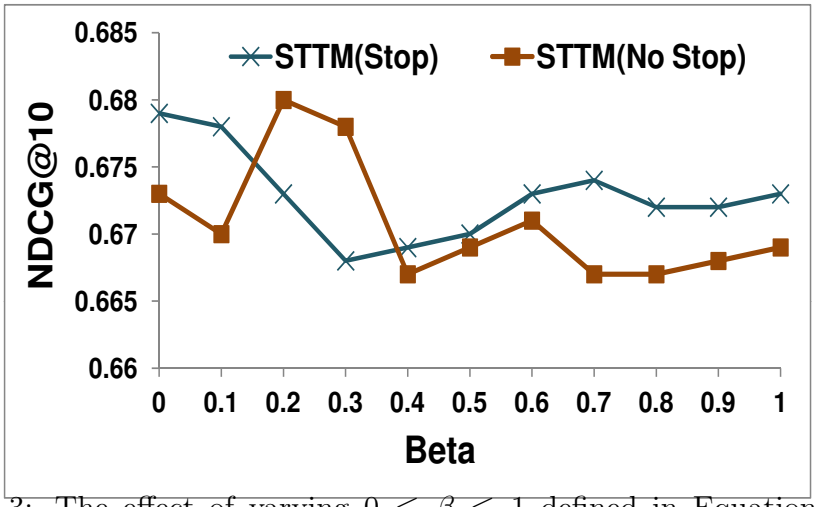


Figure 9.3: The effect of varying $0 \leq \beta \leq 1$ defined in Equation 9.12. We obtained statistically significant results according to paired t-test ($p < 0.01$) against all comparative methods.

9.3 Closing Remarks

We have presented our model `STTM` that re-ranks text documents based on conceptual difficulty. Our major innovation lies in the way we have adopted a conceptual model to solve the problem. Traditional readability formulae cannot capture domain-specific jargon, for example, "star", "shock" etc. By maintaining term order in the document, our model captures inter-segment cohesion among neighboring terms. We have also shown that stopwords play some role in determining reading difficulty of text documents.

Chapter Ten

Conclusions and Future Directions

10.1 Summary of the Methods

This thesis presented probabilistic parametric and nonparametric topic models for text data which takes into consideration word order of the document. The main motivation for incorporating word order information in a topic model is to capture the semantic or the logical storyline inherent in the document which is generally lost when we assume that the order of words in the document is not important. One of the models, NTSeg maintains the document structure such as paragraphs and sentences (in general, segments) alongwith word order in the document. The models proposed in this thesis assume a Markovian structure on the order of words and segments in the documents. The bi-gram status indicator variable enabled the n-gram topic models to generate topical n-grams or phrasal terms which lead to better topic interpretability. The temporal n-gram topic model proposed in this thesis can capture how n-gram topics have changed over time. The temporal n-gram model incorporates a time-stamp variable in the n-gram topic model and generates n-gram topical words over time. In all the models, topical phrases can be formed by concatenating the words in sequence based on the value of the bigram status variable. This thesis also described a supervised topic model with word order that considers side information during learning the parameters of the model. The side information can help the topic model learn more fine grained topics as compared to the topics generated by the unsupervised topic models,. In addition, the nonparametric topic models proposed in this thesis can automatically find out the number of latent topics that pervades in the document collection based on the characteristics of the data.

This thesis presented the terrain models to predict the readability of domain-specific documents. The models compute the reading difficulty in the latent semantic indexing space, which brings better word-document correlations. The readability model proposed in this thesis computed the readability score of a document based on the conceptual hops in sequence i.e. the model maintains the order of the words. We then used the notion of cohesion and term difficulty to compute the overall

domain-specific reading difficulty score of a document. The main innovation that the readability model proposed in this thesis is that it can differentiate the difficulty of a domain-specific concept based on context, which other readability methods cannot perform.

10.2 Shortcomings of the Models

The models proposed in this thesis indeed have many shortcomings, which would lead to some interesting future works.

1. Computational complexity of the models is a challenge, which need to be addressed. The complex graphical models with large parameter set, make the models computationally demanding.

2. The generation of phrasal terms by concatenating the bigrams in sequence is still a very simplistic solution to the problem of generation of topical n-grams. This method may not work all the time.

3. The nonparametric topic models proposed by us have a shortcoming in that one needs to store the word order information in an external matrix. This is again demanding in terms of space complexity.

4. Data sparsity might a problem sometimes, and better models to handle sparse data is needed.

5. Our supervised bigram topic model always generates bigrams which inherits the same limitations as in the BTM [252].

6. Our NTOT model inherits the limitations that are currently in the TNG.

7. Words in a phrase in all our model do not share the same probability mass.

8. We need to work towards designing better latent semantic models for readability prediction. The reason is that the latent concept model that we have used in our work has many shortcomings when applied to text data.

10.3 Suggestions for Future Research

1. The n-gram topic models are more expensive in both space and time complexity as compared to their unigram based models. It is so because these models have a large set of parameters than their unigram based models. Also, they follow the word order in the document. In addition, the input format to these models is not a traditional vector space, but the entire document with word order. If we consider smaller units of the document such as punctuation, then the resulting processing speed will be even more slower.

    - In order to speed up the inference process, we need to come up with different sampling schemes. Gibbs sampling procedure is inherently slow to converge even for a small dataset. Hence they are generally also very slow for large datasets with word order. Therefore, sampling methods such as variational methods or stochastic variational inference procedures [100] must be looked at in more detail. These methods not only scale to large scale datasets, but are very fast to converge as well. Also, they can be parallelized.

    - Another way to deal with the problem would be come up with Gibbs sampling procedures which can scale [290]. For example, we can use the samplers on clusters of computers i.e. in a distributed environment [191], [7].

2. Longer order phrase formation in our model requires concatenation of the bigram status variables, which is not a powerful way to generate phrasal terms.

Also, the probability mass between the words in a phrase is not shared, although they share the same topic. PDLDA model tried to handle exactly this issue by incorporating the HPYP model in the LDA model. But the PDLDA model ended up using the HPYP priors which also do not scale to large scale datasets.

- One way to deal with this problem is to do sampling in two stages. In the first stage, we can generate longer phrases between the words by using some phrase generation techniques. Then form those words as single chunk, and then sample those words in one topic as one single chunk. In [122], we had adopted similar approach where we generated such chunks manually, and performed concept space generation using these n-grams as one chunk. But this manual work can be done away with where the model automatically finds out appropriate phrasal terms based on the characteristics of the data.

3. Text preprocessing is also an important component in the n-gram probabilistic topic models. This is an important component that is often ignored, but it has been found during the course of the n-gram topic model development that text pre-processing is indeed an important component.

   - Stemming has shown to improve the performance of topics models.

   - It is not clear the role that the stopwords play in the n-gram topic modeling as it has been seen that they degrade the performance. In the future, we need to look the role that stopwords play. BTM model has shown the by incorporating word order, one can get rid of frequently occurring words in the latent topics. These frequently occurring words dominate the topics generated by unigram topic models.

   - Noisy terms in the document drastically deteriorate the performance of the n-gram topic models both in qualitative and quantitative analysis. In order to solve this issue, we need to adopt some aggressive techniques

to remove noisy terms in the document. In the future, we need to come up with more robust models which are susceptible to noise in text.

10.4 Personal Research Experience

The first research idea which I worked on was readability prediction. I found many problems both in the readability formulae and the fact that web search engines do not retrieve documents based on the readability of the user. Therefore, my idea was to completely digress from the current techniques and come up with something more innovative. My readability models were inspired from the Super Mario video game, which I used to play a lot when I was a child. Just like the way, Mario moves around the terrain, I thought that the same idea can also be applied for readability prediction. Thus, I used to visualize the entire document as a terrain with Super Mario trying to cross from the first word to the last. This idea was liked by many reviewers, but where the papers lacked was in evaluation as readability is still a very nascent field in information retrieval despite readability of a document in general has been around for quite a long time, since 1920s. In addition, there are no standard test collections to conduct readability evaluations.

Personally, the research experience during my doctorate study has been full of challenges. Initially, topic modeling seemed to be a bit of an intimidating field for me, given the amount of Mathematics involved in this area. But after getting into this area, and learning some of the basics, I could grasp the technicalities of topic modeling very well. I enjoy working on new probabilistic topic models, but always keep in mind how probabilistic topic models, which I design, can be applied to the readability problem and information retrieval. I sincerely hope that my work would make new and current researchers think twice before considering strong bag-of-words assumptions in their models.

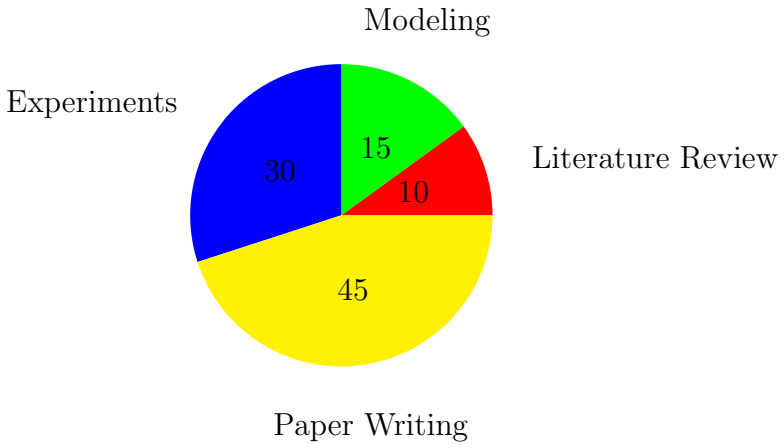


Figure 10.1: A pie chart depicting the approximate amount of time that I have spend during my research study tenure in different tasks. Note that the numbers are in percentage. Surprisingly, most of my time was spent on writing and revising the my manuscripts.

Appendix A

N-gram Topic Segmentation Model - Full Gibbs Sampling Derivation

This section shows the complete Gibbs sampling derivation of our model described in Chapter 4. In this model, our interest is in two conditional distributions, which are, $\underbrace{P(z_{si^d}, x^d_{si} | z^d_{\neg si}, x^d_{\neg si}, \mathbf{c}, \mathbf{y}, \mathbf{w})}_{\text{Word-topic distribution}}$ and $\underbrace{P(y^d_s, c^d_s | \mathbf{z}, y^d_{\neg s}, c^d_{\neg s}, \mathbf{w})}_{\text{Segment-topic distribution}}$. Exact inference is known to be intractable in the LDA family, and thus we resort to approximate inference using Gibbs sampling. This is achieved by integrating out the parameter random variables, taking advantage of conjugacy to derive a closed form for the Gibbs conditional distribution:
$P(z^d_{si}, x^d_{si}, y^d_s, c^d_s | z^d_{\neg si}, x^d_{\neg si}, y^d_{\neg s}, c^d_{\neg s}, \mathbf{w}, \beta, \alpha, \gamma, \delta, \rho, \Omega)$. As the parameter variables are integrated out during sampling, this is also known as collapsed Gibbs sampling. We start with the joint distribution:

$$P(\mathbf{z}, \mathbf{x}, \mathbf{w}, \mathbf{y}, \mathbf{c} | \beta, \alpha, \gamma, \delta, \rho, \Omega) = P(\mathbf{w}, \mathbf{x} | \mathbf{z}, \beta, \gamma, \delta) \times P(\mathbf{z} | \alpha, \mathbf{y}) \times P(\mathbf{y} | \mathbf{c}, \tau) \times P(\mathbf{c} | \Omega) \tag{A.1}$$

Each of the components can be simplified as follows:

$$
\begin{aligned}
P(\mathbf{w},\mathbf{x}|\mathbf{z},\beta,\gamma,\delta) &= \int_{\psi}\int_{\sigma}\int_{\phi}\prod_{d=1}^{D}\prod_{s=1}^{S}\prod_{n=1}^{n^{(ds)}}\Big(P(w_{si}^{d}|x_{si}^{d},\phi_{si}^{d},\sigma_{z_{si}^{d}w_{s(n-1)}^{d}}) \\
&\quad\times P(x_{si}^{d}|\psi_{z_{s(n-1)}^{d}w_{s(n-1)}^{d}})\Big)\times\prod_{l=1}^{L}\prod_{v=1}^{W}P(\sigma_{lv}|\delta)\times P(\psi_{lv}|\gamma)\mathrm{d}\Sigma\mathrm{d}\Psi \\
&\quad\times\prod_{l=1}^{L}P(\phi_{l}|\beta)\mathrm{d}\Phi \\
&= \int_{\phi}\prod_{l=1}^{L}\left[\left(\prod_{v=1}^{W}\phi_{lv}^{n_{lv}}\right)\times\frac{\Gamma\big(\sum_{v=1}^{W}\beta_{v}\big)}{\prod_{v=1}^{W}\Gamma(\beta_{v})}\prod_{v=1}^{W}\phi_{lv}^{\beta_{v}-1}\right]\mathrm{d}\Phi \\
&\quad\times\int_{\psi}\prod_{l=1}^{L}\prod_{w=1}^{W}\left[\left(\prod_{v=1}^{W}\sigma_{lwv}^{m_{lwv}}\right)\frac{\Gamma\big(\sum_{v=1}^{W}\delta_{v}\big)}{\prod_{v=1}^{W}\Gamma(\delta_{v})}\prod_{v=1}^{W}\sigma_{lwv}^{\delta_{v}-1}\right]\mathrm{d}\Sigma \\
&\quad\times\int_{\psi}\prod_{l=1}^{L}\prod_{w=1}^{W}\left[\left(\prod_{t=0}^{1}\psi_{lwt}^{p_{lwt}}\right)\frac{\Gamma\big(\sum_{t=0}^{1}\gamma_{t}\big)}{\prod_{t=0}^{1}\Gamma(\gamma_{t})}\prod_{t=0}^{1}\psi_{lwt}^{\gamma_{t}-1}\right]\mathrm{d}\Psi \\
&\quad\propto\prod_{l=1}^{L}\underbrace{\frac{\prod_{v=1}^{W}\Gamma\big(n_{lv}+\beta_{v}\big)}{\Gamma\big(\sum_{v=1}^{W}(n_{lv}+\beta_{v})\big)}}_{\text{Unigram sampling}}\prod_{l=1}^{L}\prod_{w=1}^{W}\underbrace{\frac{\prod_{v=1}^{W}\Gamma(m_{lwv}+\delta_{v})}{\Gamma\big(\sum_{v=1}^{W}(m_{lwv}+\delta_{v})\big)}}_{\text{Bigram sampling}} \\
&\quad\prod_{l=1}^{L}\prod_{w=1}^{W}\underbrace{\frac{\prod_{t=0}^{1}\Gamma(p_{lwt}+\gamma_{t})}{\Gamma\big(\sum_{t=0}^{1}(p_{lwt}+\gamma_{t})\big)}}_{\text{Bigram status sampling}}
\end{aligned}
\tag{A.2}
$$

$$
\begin{aligned}
P(\mathbf{z}|\alpha,\mathbf{y}) &= \int_{\theta}P(\mathbf{z}|\theta)P(\theta|\alpha,\mathbf{y})\mathrm{d}\theta \\
&= \int_{\theta}\prod_{d=1}^{D}\prod_{s=1}^{S}\left(\prod_{n=1}^{n^{(ds)}}P(z_{si}^{d}|\theta^{(s)})P(\theta^{(s)}|\alpha,y_{s}^{d})\right)\mathrm{d}\Theta \\
&= \int_{\theta}\prod_{d=1}^{D}\prod_{s=1}^{S}\prod_{l=1}^{L}\theta_{n_{l}^{(ds)}}^{sl}\prod_{d=1}^{D}\prod_{s=1}^{S}\left(\frac{\Gamma\big(\sum_{l=1}^{L}\alpha_{y_{s}^{d}l}\big)}{\prod_{l=1}^{L}\Gamma(\alpha_{y_{s}^{d}l})}\prod_{l=1}^{L}\theta_{\alpha_{y_{s}^{d}l}-1}^{(sl)}\right)\mathrm{d}\Theta \\
&= \prod_{d=1}^{D}\prod_{s=1}^{S}\Bigg(\underbrace{\frac{\Gamma\big(\sum_{l=1}^{L}\alpha_{y_{s}^{d}l}\big)}{\prod_{l=1}^{L}\Gamma(\alpha_{y_{s}^{d}l})}}_{\substack{\text{Segment and word}\\ \text{topic proportion sampling}}}\Bigg)\prod_{d=1}^{D}\prod_{s=1}^{S}\frac{\prod_{l=1}^{L}\Gamma\big(\alpha_{y_{s}^{d}l}+n_{l}^{(ds)}\big)}{\Gamma\big(\sum_{l=1}^{L}\alpha_{y_{s}^{d}l}+n_{l}^{(ds)}\big)}
\end{aligned}
\tag{A.3}
$$

$$
\begin{aligned}
P(\mathbf{y}|\mathbf{c},\delta) &= \int_{\tau} P(\mathbf{y}|\tau,\mathbf{c})P(\tau|\rho)\mathrm{d}\tau \\
&= \int_{\tau} \prod_{d=1}^{D} \left(\prod_{s=1}^{S} P(y_s^d|\tau_d, c_s^d) P(\tau_d|\rho) \right) \mathrm{d}\tau \\
&= \int_{\tau} \left(\prod_{u\in U_1} \prod_{s=1}^{|S_u|} P(y_s^d|\tau_d) \frac{\Gamma\big(\sum_{k=1}^{K}\delta_k\big)}{\prod_{k=1}^{K}\Gamma(\rho_k)} \prod_{k=1}^{K} \tau_{dk}^{\rho_k - 1} \right) \mathrm{d}\tau \\
&= \prod_{d=1}^{D} \prod_{u\in U_1} \left(\frac{\Gamma\big(\sum_{k=1}^{K}\rho_k\big)}{\prod_{k=1}^{K}\Gamma(\rho_k)} \right) \prod_{d=1}^{D} \prod_{u\in U_1} \underbrace{\frac{\prod_{k=1}^{K}\Gamma\big(\rho_k + n_k^{(S_u)}\big)}{\Gamma\big(\sum_{k=1}^{K}\rho_k + n^{(S_u)}\big)}}_{\text{Segment sampling}}
\end{aligned} \tag{A.4}
$$

$$
\begin{aligned}
P(\mathbf{c}|\Omega) &= \int_{0}^{1} P(\mathbf{c}|\pi)P(\pi)\mathrm{d}\Pi \\
&= \int_{0}^{1} \prod_{d=1}^{D} \left(\prod_{s=1}^{S} P(c_s^d|\pi_d) P(\pi_d|\Omega) \right) \mathrm{d}\Pi \\
&= \frac{\Gamma(2\Omega)}{\Gamma(\Omega)^2} \times \underbrace{\frac{\Gamma(N_s^d + 2\Omega)}{\Gamma(n_1+\Omega)\Gamma(n_o+\Omega)}}_{\text{Segment status sampling}}
\end{aligned} \tag{A.5}
$$

Using the chain and $\Gamma(\alpha) = (\alpha-1)\Gamma(\alpha-1)$, we can obtain the conditional probability conveniently,

$$
\begin{aligned}
&P(z_{si}^d, x_{si}^d|\mathbf{w}, z_{\neg si}^d, x_{\neg si}^d, \mathbf{y}, \alpha, \beta, \gamma, \delta) = \\
&\frac{P(w_{si}^d, z_{si}^d, x_{si}^d|w_{\neg si}^d, z_{\neg si}^d, x_{\neg si}^d, \mathbf{y}, \alpha, \beta, \gamma, \delta)}{P(w_{si}^d|w_{\neg si}^d, z_{\neg si}^d, x_{\neg si}^d, \mathbf{y}, \alpha, \beta, \gamma, \delta)}
\end{aligned} \tag{A.6}
$$

$$
\begin{aligned}
&\propto (\alpha_{y_s^d z_{si}^d} + n_{z_{si}^d}^{(ds)} - 1) \times (\gamma_{x_{si}^d} + p_{z_{s(n-1)}^d w_{s(n-1)}^d x_{si}^d} - 1) \\
&\times \begin{cases} \frac{\beta_{w_{si}^d} + n_{z_{si}^d w_{si}^d} - 1}{\sum_{v=1}^{W}\left(\beta_v + n_{z_{si}^d v}\right) - 1} & \text{if } x_{si}^d = 0 \\ \frac{\delta_{w_{si}^d} + m_{z_{si}^d w_{s(n-1)}^d w_{si}^d} - 1}{\sum_{v=1}^{W}\left(\delta_v + m_{z_{si}^d w_{s(n-1)}^d v}\right) - 1} & \text{if } x_{si}^d = 1 \end{cases}
\end{aligned} \tag{A.7}
$$

Again, using the chain rule we obtain:

$$P(y_s^d, c_s^d|\mathbf{z}, y_{\neg s}^d, c_{\neg s}^d, \mathbf{w}) = \frac{P(\mathbf{z}, \mathbf{x}, \mathbf{y}, \mathbf{c}, \mathbf{w})}{P(\mathbf{z}, \mathbf{x}, y_{\neg s}^d, c_{\neg s}^d, \mathbf{w})} \propto \frac{P(\mathbf{z}|\mathbf{y}), P(\mathbf{y}|\mathbf{c}), P(\mathbf{c})}{P(z_{\neg s}|y_{\neg s}^d), P(y_{\neg s}^d|c_{\neg s}^d), P(c_{\neg s}^d)} \tag{A.8}$$

If $c_s^d = 0$, then we obtain the following:

$$P(y_s^d, c_s^d|\mathbf{z}, y_{\neg s}^d, c_{\neg s}^d \mathbf{w}) = \left(\frac{\Gamma\big(\sum_{l=1}^{L} \alpha_{y_s^d l}\big)}{\prod_{l=1}^{L} \Gamma(\alpha_{y_s^d l})}\right) \times \left(\frac{\prod_{l=1}^{L} \Gamma\big(\alpha_{y_s^d l} + n_l^{(ds)}\big)}{\Gamma\big(\sum_{l=1}^{L} \alpha_{y_s^d l} + n_l^{(ds)}\big)}\right) \left(\frac{\prod_{k=1}^{K} \Gamma\big(\rho_k + n_k^{(S_u^0)}\big)}{\Gamma\big(\sum_{k=1}^{K} \rho_k + n_0^{(S_u^0)}\big)}\right) \times \left(\frac{n_0 + \Omega}{N_s^d + 2\Omega}\right) \tag{A.9}$$

If $c_s^d = 1$, then we obtain:

$$P(y_s^d, c_s^d|\mathbf{z}, y_{\neg s}^d, c_{\neg s}^d \mathbf{w}) = \left(\frac{\Gamma\big(\sum_{l=1}^{L} \alpha_{y_s^d l}\big)}{\prod_{l=1}^{L} \Gamma(\alpha_{y_s^d l})}\right) \times \left(\frac{\prod_{l=1}^{L} \Gamma\big(\alpha_{y_s^d l} + n_l^{(ds)}\big)}{\Gamma\big(\sum_{l=1}^{L} \alpha_{y_s^d l} + n_l^{(ds)}\big)}\right) \left(\frac{\Gamma\big(\sum_{k=1}^{K} \rho_k\big)}{\prod_{k=1}^{K} \Gamma(\rho_k)}\right) \times \left(\frac{\prod_{k=1}^{K} \Gamma\big(\rho_k + n_k^{(S_{u-1}^1)}\big)}{\Gamma\big(\sum_{k=1}^{K} \rho_k + n_1^{(S_{u-1}^1)}\big)}\right) \left(\frac{\prod_{k=1}^{K} \Gamma\big(\rho_k + n_k^{(S_u^1)}\big)}{\Gamma\big(\sum_{k=1}^{K} \rho_k + n_1^{(S_u^1)}\big)}\right) \times \left(\frac{n_1 + \Omega}{N_s^d + 2\Omega}\right) \tag{A.10}$$

Appendix B

N-gram Topics Over Time Model - Full Gibbs Sampling Derivation

The complete Gibbs sampling derivation in this section corresponds to the model proposed in Chapter 5. In this model, in order to derive the collapsed Gibbs sampling procedure for the n-gram topic-over time model, we first begin with the joint distribution:

$$P(\mathbf{w}, \mathbf{x}, \mathbf{t}, \mathbf{z}|\alpha, \beta, \gamma, \delta, \Omega) \tag{B.1}$$

In order to simplify the integration process, the property of conjugate priors will be helpful. We start with the joint distribution:

$$\underbrace{P(\mathbf{w}, \mathbf{x}, \mathbf{t}, \mathbf{z}|\alpha, \beta, \gamma, \delta, \Omega)}_{\text{Full joint probability}} = \underbrace{P(\mathbf{w}, \mathbf{x}|\mathbf{z}, \beta, \delta, \gamma)}_{\text{Word and bigram status probability}} \times \underbrace{P(\mathbf{t}|\mathbf{z}, \Omega)}_{\text{Temporal probability}} \times \underbrace{P(\mathbf{z}|\alpha)}_{\text{Topic probability}} \tag{B.2}$$

Each of the components can be simplified as follows:

$$
\begin{aligned}
P(\mathbf{w},\mathbf{z},\mathbf{x}|\beta,\delta,\gamma) &= \int_{\psi}\int_{\sigma}\int_{\phi}\prod_{d=1}^{D}\prod_{i=1}^{N^d}\left(P(w_i^d|x_i^d,\phi_{z_i}^d,\sigma_{z_i{}^d w_{i-1}^d})\times P(x_i^d|\psi_{z_{i-1}^d w_{i-1}^d})\right)\\
&\quad\prod_{z=1}^{L}\prod_{v=1}^{W}P(\sigma_{zv}|\delta)P(\psi_{zv}|\gamma)\mathrm{d}\Sigma\mathrm{d}\Psi\prod_{z=1}^{L}P(\phi_z|\beta)\mathrm{d}\Phi\\
&= \int_{\phi}\prod_{z=1}^{L}\left(\prod_{v=1}^{W}\phi_{zv}^{n_{zv}}\frac{\Gamma\left(\sum_{v=1}^{W}\beta_v\right)}{\prod_{v=1}^{W}\Gamma(\beta_v)}\prod_{v=1}^{W}\phi_{zv}^{\beta_v-1}\right)\mathrm{d}\Phi\\
&\quad\times\int_{\sigma}\prod_{z=1}^{L}\prod_{w=1}^{W}\left(\prod_{v=1}^{W}\sigma_{zwv}^{m_{zwv}}\frac{\Gamma\left(\sum_{v=1}^{W}\delta_v\right)}{\prod_{v=1}^{W}\Gamma(\delta_v)}\prod_{v=1}^{W}\sigma_{zwv}^{\delta_v-1}\right)\mathrm{d}\Sigma\\
&\quad\times\int_{\psi}\prod_{z=1}^{L}\prod_{w=1}^{W}\left(\prod_{k=0}^{1}\psi_{zwk}^{p_{zwk}}\frac{\Gamma\left(\sum_{k=0}^{1}\gamma_v\right)}{\prod_{k=0}^{1}\Gamma(\gamma_k)}\prod_{k=0}^{1}\psi_{zwk}^{\gamma_k-1}\right)\mathrm{d}\Psi\\
&\propto \prod_{z=1}^{L}\underbrace{\frac{\prod_{v=1}^{W}\Gamma\left(n_{zv}+\beta_v\right)}{\Gamma\left(\sum_{v=1}^{W}\left(n_{zv}+\beta_v\right)\right)}}_{\text{Unigram sampling}}\prod_{z=1}^{L}\prod_{w=1}^{W}\underbrace{\frac{\prod_{v=1}^{W}\Gamma\left(m_{zwv}+\delta_v\right)}{\Gamma\left(\sum_{v=1}^{W}\left(m_{zwv}+\delta_v\right)\right)}}_{\text{Bigram sampling}}\\
&\quad\prod_{z=1}^{L}\prod_{w=1}^{W}\underbrace{\frac{\prod_{k=0}^{1}\Gamma\left(p_{zwk}+\gamma_k\right)}{\Gamma\left(\sum_{k=0}^{1}\left(p_{zwk}+\gamma_k\right)\right)}}_{\text{Bigram status sampling}}
\end{aligned}
\tag{B.3}
$$

$$
P(\mathbf{t}|\mathbf{z},\Omega)=\prod_{d=1}^{D}\prod_{i=1}^{N^d}P(t_i^d|\Omega_{z_i^d})
\tag{B.4}
$$

$$
\begin{aligned}
P(\mathbf{z}|\alpha) &= \int_{\theta} \prod_{d=1}^{D} \left(\prod_{i=1}^{N^d} P(z_i^d|\theta_d) P(\theta_d|\alpha) \right) \\
&= \int_{\theta} \prod_{d=1}^{D} \left(\prod_{z=1}^{L} \theta_{dz}^{q_{dz}} \frac{\Gamma\big(\sum_{z=1}^{L} \alpha_z\big)}{\prod_{z=1}^{L}(\alpha_z)} \prod_{z=1}^{L} \theta_{dz}^{\alpha_z - 1} \right) \mathrm{d}\Theta \\
&\propto \prod_{d=1}^{D} \frac{\prod_{z=1}^{L} \Gamma\big(q_{dz} + \alpha_z\big)}{\Gamma\big(\sum_{z=1}^{L}(q_{dz} + \alpha_z)\big)}
\end{aligned}
\tag{B.5}
$$

Using the chain rule and $\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$, we can obtain the conditional probability conveniently:

$$
\begin{aligned}
P(z_i^d, x_i^d|\mathbf{w}, \mathbf{t}, z_i^d, x_i^d, \alpha, \beta, \gamma, \delta, \Omega) &= \frac{P(z_i^d, w_i^d, x_i^d, t_i^d|w_{\neg i}^d, t_{\neg i}^d, z_{\neg i}^d, \alpha, \beta, \gamma, \delta, \Omega)}{P(w_i^d, t_i^d|w_{\neg i}^d, x_{\neg i}^d, t_{\neg i}^d, z_{\neg i}^d, \alpha, \beta, \gamma, \delta, \Omega)} \\
&\propto \frac{P(\mathbf{w}, \mathbf{x}, \mathbf{t}, \mathbf{z}|\alpha, \beta, \gamma, \delta, \Omega)}{P(w_{\neg i}^d, t_{\neg i}^d, z_{\neg i}^d, x_{\neg i}^d|\alpha, \beta, \gamma, \delta, \Omega)}
\end{aligned}
\tag{B.6}
$$

$$
\propto \big(\gamma_{x_i{}^d} + p_{z_{i-1}^d w_{i-1}^d} - 1\big)\big(\alpha_{z_i^d} + q_{dz_i^d} - 1\big) \times \underbrace{\frac{(1 - t_i^d)^{\Omega_{z_i^d 1} - 1} t_i^{d\Omega_{z_i^d 2} - 1}}{B(\Omega_{z_i^d 1}, \Omega_{z_i^d 2})}}_{\text{Temporal information sampling}} \times
$$

$$
\begin{cases}
\frac{\beta_{w_i^d} + n_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W}(\beta_v + n_{z_i^d v}) - 1} & \text{if } x_i^d = 0 \\
\frac{\delta_{w_i^d} + m_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W}(\delta_v + m_{z_i^d w_{i-1}^d}) - 1} & \text{if } x_i^d = 1
\end{cases}
\tag{B.7}
$$

Appendix C

Bigram Topic Model - Full Gibbs Sampling Derivation

Computing the exact posterior estimate in the BTM model [252] is intractable due to integration over a large state space. Therefore, we resort to approximation techniques such as Gibbs sampling [78]. Other estimation techniques such as EM [59] has problems related to getting stuck in the local optima and some tempered algorithms [101, 102] have been proposed to smooth the parameters of the model for acceptable predictive performance, but still they do not solve the problem of over-fitting and local optima [23, 207]. We show here the Gibbs sampling for computing the posterior inference for the BTM. The training data here is the complete document with word order kept intact. We begin with the joint distribution $P(\mathbf{w}, \mathbf{z}|\alpha, \beta)$. In order to simplify the integrals, we can take advantage of conjugate priors[1]. Note that all the counts used below exclude the current case i.e. the word being visited during sampling. When we use a $\neg$ sign in the subscript of a variable it means that the variable corresponding to the subscripted index is remove from the calculation of the count. Variables in bold, for example, $\mathbf{w}$ is defined as $w_i^d : \forall d, i$.

$$
\begin{aligned}
P(\mathbf{w}, \mathbf{z}|\alpha, \beta) &= \int \prod_{d=1}^{D} \prod_{i=1}^{N^d} P(w_i^d|\phi_{z_i^d w_{i-1}^d}) \prod_{z=1}^{L} \left(\prod_{v=1}^{W} P(\phi_{zv}|\beta) \right) \mathrm{d}\Phi \int \prod_{d=1}^{D} \left(\prod_{i=1}^{N^d} P(z_i^d|\theta^d) P(\theta^d|\alpha) \right) \mathrm{d}\Theta \\
&= \int \prod_{z=1}^{L} \prod_{w=1}^{W} \left(\prod_{v=1}^{W} \phi_{zwv}^{m_{zwv}} \frac{\Gamma\left(\sum_{v=1}^{W} \beta_v\right)}{\prod_{v=1}^{W} \Gamma(\beta_v)} \prod_{v=1}^{W} \phi_{zwv}^{\beta_v - 1} \right) \mathrm{d}\Phi \times \int \prod_{d=1}^{D} \left(\prod_{z=1}^{L} \theta_{dz}^{q_{dz}} \frac{\Gamma\left(\sum_{z=1}^{L} \alpha_z\right)}{\prod_{z=1}^{L} \Gamma(\alpha_z)} \prod_{z=1}^{T} \theta_{dz}^{\alpha_z - 1} \right) \mathrm{d}\Theta \\
&\propto \underbrace{\prod_{z=1}^{L} \prod_{w=1}^{W} \frac{\prod_{v=1}^{W} \Gamma(m_{zwv} + \beta_v)}{\Gamma(\sum_{v=1}^{W} (m_{zwv} + \beta_v))}}_{\text{Probability of bigram in a topic}} \quad \underbrace{\prod_{d=1}^{D} \frac{\prod_{z=1}^{L} \Gamma(q_{dz} + \alpha_z)}{\Gamma(\sum_{z=1}^{L} (q_{dz} + \alpha_z))}}_{\text{Probability of a topic in a document}}
\end{aligned}
\tag{C.1}
$$

We know that $\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$, so now we can obtain the following conditional probability distribution:

[1] We have shown the derivation of the model using conjugate and symmetric priors.

$$P(z_i^d|\mathbf{w}, \mathbf{z}_{\neg i}^d, \alpha, \beta) = \frac{P(w_i^d, z_i^d|\mathbf{w}_{\neg i}^d, \mathbf{z}_{\neg i}^d, \mathbf{z}_{\neg i}^d, \alpha, \beta)}{P(w_i^d|\mathbf{w}_{\neg i}^d, \mathbf{z}_{\neg i}^d, \alpha, \beta)}$$
$$= \left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{L} \left(\alpha_z + q_{dz}\right) - 1} \right) \times \left(\frac{\beta_{w_i^d} + m_{z_i^d w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W} \left(\beta_v + m_{z_i^d w_{i-1}^d v}\right) - 1} \right) \tag{C.2}$$

Equivalently, the expression above can be written as:

$$P(z_i^d|\mathbf{w}, \mathbf{z}_{\neg i}^d, \alpha, \beta) = \underbrace{\left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{L} \left(\alpha_z + q_{dz}\right) - 1} \right)}_{\text{Same as the LDA model}} \times \underbrace{\left(\frac{\beta_{w_i^d} + m_{w_{i-1}^d w_i^d z_i^d} - 1}{\sum_{v=1}^{W} (\beta_v + m_{z_i^d w_{i-1}^d v}) - 1} \right)}_{\text{Bigrams in topics}} \tag{C.3}$$

The posterior estimates of θ, ϕ can be written as follows:

$$\hat{\theta}_z^d = \underbrace{\frac{\alpha_z + q_{dz}}{\sum_{t=1}^{L} (\alpha_t + q_{dt})}}_{\substack{\text{Posterior for} \\ \text{document topic} \\ \text{proportions}}} \tag{C.4}$$

$$\hat{\phi}_{zw} = \underbrace{\frac{\beta_w + m_{wvz}}{\sum_{v=1}^{W} (\beta_v + m_{wvz})}}_{\substack{\text{Posterior for} \\ \text{bigram topic} \\ \text{proportions}}} \tag{C.5}$$

Appendix D

LDA-Collocation Model - Full Gibbs Sampling Derivation

The complete derivation of the Gibbs sampling for the LDACOL model is as follows:

$$
\begin{aligned}
P(\mathbf{w},\mathbf{x},\mathbf{z}|\alpha,\beta,\gamma,\delta) &= \int\int\int \prod_{d=1}^{D}\prod_{i=1}^{N^d}\Big(P(w_i^d|x_i^d,\phi_{z_i^d},\sigma_{w_{i-1}^d})P(x_i^d|\psi_{w_{i-1}^d})\Big)\prod_{v=1}^{W}P(\sigma_v|\delta)P(\psi_v|\gamma)\mathrm{d}\Sigma\mathrm{d}\Psi \\
&\qquad \prod_{z=1}^{L}P(\phi_z|\beta)\mathrm{d}\Phi\int\prod_{d=1}^{D}\left(\prod_{i=1}^{N^d}P(z_i^d|\theta^d)P(\theta^d|\alpha)\right)\mathrm{d}\Theta \\
&= \int\prod_{z=1}^{L}\left(\prod_{v=1}^{W}\phi_{zv}^{n_{zv}}\frac{\Gamma\Big(\sum_{v=1}^{W}\beta_v\Big)}{\prod_{v=1}^{W}\Gamma(\beta_v)}\prod_{v=1}^{W}\phi_{zv}^{\beta_v-1}\right)\mathrm{d}\Phi\times\int\prod_{w=1}^{V}\left(\prod_{v=1}^{W}\sigma_{wv}^{m_{mv}}\frac{\Gamma\Big(\sum_{v=1}^{W}\delta_v\Big)}{\prod_{v=1}^{W}\Gamma(\delta_v)}\prod_{v=1}^{W}\sigma_{wv}^{\delta_v-1}\right)\mathrm{d}\Sigma \\
&\quad\times\int\prod_{w=1}^{W}\left(\prod_{s=0}^{1}\psi_{ws}^{p_{ws}}\frac{\Gamma\Big(\sum_{s=0}^{1}\gamma_v\Big)}{\prod_{s=0}^{1}\Gamma(\gamma_s)}\prod_{s=0}^{1}\psi_{ws}^{\gamma_s-1}\right)\mathrm{d}\Psi\times\int\prod_{d=1}^{D}\left(\prod_{z=1}^{L}\theta_{dz}^{q_{dz}}\frac{\Gamma\Big(\sum_{z=1}^{L}\alpha_z\Big)}{\prod_{z=1}^{L}\Gamma(\alpha_z)}\prod_{z=1}^{T}\theta_{dz}^{\alpha_z-1}\right)\mathrm{d}\Theta \\
&\propto \underbrace{\prod_{z=1}^{L}\frac{\prod_{v=1}^{W}\Gamma(n_{zv}+\beta_v)}{\Gamma(\sum_{v=1}^{W}(n_{zv}+\beta_v))}}_{\text{Same as LDA}}\underbrace{\prod_{w=1}^{W}\frac{\prod_{v=1}^{W}\Gamma(m_{mv}+\delta_v)}{\Gamma(\sum_{v=1}^{W}(m_{wv}+\delta_v))}}_{\text{Capture collocations}}\underbrace{\prod_{w=1}^{W}\frac{\prod_{s=0}^{1}\Gamma(p_{ws}+\gamma_s)}{\Gamma(\sum_{s=0}^{1}(p_{ws}+\gamma_s))}}_{\text{Bigram status indicator}}\underbrace{\prod_{d=1}^{D}\frac{\prod_{z=1}^{L}\Gamma(q_{dz}+\alpha_z)}{\Gamma(\sum_{z=1}^{L}(q_{dz}+\alpha_z))}}_{\text{As in LDA}}
\end{aligned}
$$

We know that $\Gamma(\alpha)=(\alpha-1)\Gamma(\alpha-1)$, so now we can obtain the following conditional probability distribution:

$$
\begin{aligned}
P(z_i^d,x_i^d|\mathbf{w},\mathbf{z}_{\neg i}^d,\mathbf{x}_{\neg i}^d,\alpha,\beta,\gamma,\delta) &= \frac{P(w_i^d,z_i^d,x_i^d|\mathbf{w}_{\neg i}^d,\mathbf{z}_{\neg i}^d,\mathbf{z}_{\neg i}^d,\alpha,\beta,\gamma,\delta)}{P(w_i^d|\mathbf{w}_{\neg i}^d,\mathbf{x}_{\neg i}^d,\mathbf{z}_{\neg i}^d,\alpha,\beta,\gamma,\delta)} \\
&= \left(\frac{\gamma_{x_i^d}+p_{w_{i-1}^d x_i^d}-1}{\sum_{s=0}^{1}\left(\gamma_s+p_{w_{i-1}^d s}\right)-1}\right)\left(\frac{\alpha_{z_i^d}+q_{dz_i^d}-1}{\sum_{z=1}^{L}\left(\alpha_z+q_{dz}\right)-1}\right)\times\begin{cases}\frac{\beta_{w_i^d}+n_{z_i^d w_i^d}-1}{\sum_{v=1}^{W}(\beta_v+n_{z_i^d v})-1} & \text{if } x_i^d=0\\ \frac{\delta_{w_i^d}+m_{w_{i-1}^d w_i^d}-1}{\sum_{v=1}^{W}(\delta_v+m_{w_{i-1}^d v})-1} & \text{if } x_i^d=1\end{cases}
\end{aligned}
\tag{D.1}
$$

Equivalently, the expression above can be written as:

$$P(z_i^d|\mathbf{w}, \mathbf{z}_{\neg i}^d, \mathbf{x}, \alpha, \beta, \gamma, \delta) = \begin{cases} \underbrace{\left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{L}\left(\alpha_z + q_{dz}\right) - 1}\right) \times \frac{\beta_{w_i^d} + n_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W}(\beta_v + n_{z_i^d v}) - 1}}_{\text{Same as in LDA}} & \text{if } x_i^d = 0 \\ \left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{L}\left(\alpha_z + q_{dz}\right) - 1}\right) \times \frac{\delta_{w_i^d} + m_{w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W}(\delta_v + m_{w_{i-1}^d v}) - 1} & \text{if } x_i^d = 1 \end{cases} \tag{D.2}$$

and,

$$P(x_i^d|\mathbf{w}, \mathbf{z}, \mathbf{x}_{\neg i}^d, \alpha, \beta, \gamma, \delta) = \underbrace{\left(\frac{\gamma_{x_i^d} + p_{w_{i-1}^d x_i^d} - 1}{\sum_{s=0}^{1}\left(\gamma_s + p_{w_{i-1}^d s}\right) - 1}\right)}_{\text{Sample bigram status}} \times \begin{cases} \frac{\beta_{w_i^d} + n_{z_i^d w_i^d} - 1}{\sum_{v=1}^{W}(\beta_v + n_{z_i^d v}) - 1} & \text{if } x_i^d = 0 \\ \frac{\delta_{w_i^d} + m_{w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W}(\delta_v + m_{w_{i-1}^d v}) - 1} & \text{if } x_i^d = 1 \end{cases} \tag{D.3}$$

The posterior estimates of $\theta, \phi, \psi, \sigma$ can be written as follows:

$$\hat{\theta}_z^d = \frac{\alpha_z + q_{dz}}{\sum_{t=1}^{L}(\alpha_t + q_{dt})} \tag{D.4}$$

$$\hat{\phi}_{zw} = \frac{\beta_w + n_{zw}}{\sum_{v=1}^{W}(\beta_v + n_{zv})} \tag{D.5}$$

$$\hat{\sigma}_{wv} = \frac{\delta_v + m_{wv}}{\sum_{v=1}^{W}(\delta_v + m_{wv})} \tag{D.6}$$

$$\hat{\psi}_{ws} = \frac{\gamma_k + p_{ws}}{\sum_{s=0}^{1}(\gamma_s + p_{ws})} \tag{D.7}$$

Appendix E

Proof of Bayes' Theorem when Cast into an Optimization Problem

Proof for Equation 3.46

From Equation 3.45 based on the formula of Bayes' Theorem, we can deduce that $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \alpha, \beta)$ is the posterior distribution that needs to be found out. $P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ is the prior distribution, and the denominator $P(\boldsymbol{W}|\alpha, \beta)$ is the marginal distribution over data.

We know that the Kullback-Leibler Divergence (KLD) from a distribution p to a distribution q can be written as KLD$(q||p)$. Suppose we consider an arbitrary distribution $Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \alpha, \beta)$. Our goal is to ensure that this distribution is equal to the posterior distribution $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \alpha, \beta)$. As in the Bayes' rule, this posterior is obtained by iteratively updating the prior $P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$.

Suppose we want to minimize the divergence between the arbitrary distribution and the posterior distribution, and this is what we want to achieve so that the two distributions are as close as possible or equal to each other i.e. they overlap. We can write the statement mathematically as:

$$\underset{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}}\,\text{KLD}[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] \tag{E.1}$$

We know from Equation 3.45 that:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \alpha, \beta) = \frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\alpha, \beta)} \tag{E.2}$$

So, replacing $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{W}, \alpha, \beta)$ with the right hand side formulation in Equation E.1, we obtain:

$$\underset{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}}\,\text{KLD}\left[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||\frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\alpha, \beta)}\right] \tag{E.3}$$

This equation equivalently can be written as:

$$\mathbb{E}_Q\left[\ln \frac{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{\frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\alpha, \beta)}}\right] \tag{E.4}$$

The above formulation can now be written as:

$$\mathbb{E}_Q\left[\ln \frac{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)} - \ln P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) + \ln P(\boldsymbol{W}|\alpha, \beta)\right] \tag{E.5}$$

This can be further written as:

$$\underset{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \text{KLD}\left[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||\frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{W}|\alpha, \beta)}\right] \tag{E.6}$$

This now simplifies to:

$$\underset{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \text{KLD}(Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)) - \mathbb{E}_{\mathbb{Q}}[\ln P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] + \ln P(\boldsymbol{W}|\alpha, \beta) \tag{E.7}$$

We can in-fact drop the last term in Equation E.7 because it does not depend on $\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}$, so we get:

$$\underset{Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}) \in \mathbb{P}}{\text{minimize}} \text{KLD}(Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)) - \mathbb{E}_Q[\ln P(\boldsymbol{W}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] \tag{E.8}$$

Appendix F

MedLDA Model - Full Collapsed Gibbs Sampling Derivation

In order proceed with the derivation of the collapsed Gibbs sampling scheme for the `MedLDA` model, we first need to define a joint distribution for words and the topics alongwith the regularization effects due to the maximum-margin posterior constraints. This joint distribution is written as:

$$P(\boldsymbol{Z}, \boldsymbol{B}|\alpha, \beta) = P(\boldsymbol{B}|\boldsymbol{Z}, \beta) \times P(\boldsymbol{Z}|\alpha) \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \tag{F.1}$$

Now we can separately calculate $P(\boldsymbol{W}, \boldsymbol{Z}, \beta)$ and $P(\boldsymbol{Z}|\alpha)$ because each of them depends on $\boldsymbol{\Phi}$ and $\boldsymbol{\Theta}$ respectively. According to the paradigm used in the `LDA` model, we can write the following:

$$P(\boldsymbol{W}|\boldsymbol{Z}, \beta) = \int P(\boldsymbol{W}|\boldsymbol{Z}, \boldsymbol{\Phi}) \times P(\boldsymbol{\Phi}|\beta) \mathrm{d}\boldsymbol{\Phi} \tag{F.2}$$

$P(\boldsymbol{\Phi}|\beta)$ is characterized by the Dirichlet distribution, so we can expand $P(\boldsymbol{\Phi}|\beta)$ as:

$$\begin{aligned} P(\boldsymbol{\Phi}|\beta) &= \prod_{k=1}^{L} P(\boldsymbol{\phi}_k|\beta) \\ &= \prod_{k=1}^{L} \frac{\Gamma(\sum_{i=1}^{|\beta|} \beta_i)}{\prod_{i=1}^{|\beta|} \Gamma(\beta_i)} \prod_{v=1}^{V} \phi_{k,v}^{\beta_v - 1} \end{aligned} \tag{F.3}$$

and, $P(\boldsymbol{W}|\boldsymbol{Z}, \boldsymbol{\Phi})$ is a multinomial distribution which can be written as:

$$\begin{aligned} P(\boldsymbol{W}|\boldsymbol{Z}, \boldsymbol{\Phi}) &= \prod_{i=1}^{W} \phi_{z_i^d, w_i^d} \\ &= \prod_{k=1}^{L} \prod_{v=1}^{V} \phi_{k,v}^{M_k^v} \end{aligned} \tag{F.4}$$

where M is the $L \times V$ count matrix which is a low-dimensional representation of the words in the collection. M_k^v is the number of times a topic k is assigned to word w_i^d.

Based on the equations derived above:

$$P(\boldsymbol{W}|\boldsymbol{Z},\beta) = \int \prod_{k=1}^{L} \prod_{k=1}^{L} \frac{\Gamma(\sum_{i=1}^{|\beta|} \beta_i)}{\prod_{i=1}^{|\beta|} \Gamma(\beta_i)} \prod_{v=1}^{V} \phi_{k,v}^{M_k^v + \beta_v - 1} \mathrm{d}\phi_k \tag{F.5}$$

Using the following property:

$$\int \prod_{k=1}^{L} f_k(\phi_k) \mathrm{d}\phi_1, \mathrm{d}\phi_2, \ldots, \mathrm{d}\phi_k = \prod_{k=1}^{L} \int f_k(\phi_k) \mathrm{d}\phi_k \tag{F.6}$$

So now we proceed as follows, we use Equation F.6:

$$\begin{aligned} P(\boldsymbol{W}|\boldsymbol{Z},\beta) &= \prod_{k=1}^{L} \left(\int \prod_{k=1}^{L} \frac{\Gamma(\sum_{i=1}^{|\beta|} \beta_i)}{\prod_{i=1}^{|\beta|} \Gamma(\beta_i)} \prod_{v=1}^{V} \phi_{k,v}^{M_k^v + \beta_v - 1} \mathrm{d}\phi_k \right) \\ &= \prod_{k=1}^{L} \left(\frac{\Gamma(\sum_{i=1}^{|\beta|} \beta_i)}{\prod_{i=1}^{|\beta|} \Gamma(\beta_i)} \int \prod_{v=1}^{V} \phi_{k,v}^{M_k^v + \beta_v - 1} \mathrm{d}\phi_k \right) \end{aligned} \tag{F.7}$$

The integration above can be computed in a closed form as:

$$P(\boldsymbol{W}|\boldsymbol{Z},\beta) = \prod_{k=1}^{L} \frac{\frac{\prod_{i=1}^{|M_k + \beta|} \Gamma(M_k + \beta_i)}{\Gamma(\sum_{i=1}^{|M_k + \beta|} (M_k + \beta_i))}}{\frac{\prod_{i=1}^{|\beta|} \Gamma(\beta_i)}{\Gamma(\sum_{i=1}^{|\beta|} \beta_i)}} \tag{F.8}$$

where M_k is the k^{th} row of matrix M.

Next we consider $P(\boldsymbol{\theta}^d|\alpha)$ which also includes a regularization effect, and we

proceed as follows:

$$
\begin{aligned}
P(\boldsymbol{\Theta}|\alpha) \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} &= \prod_{d=1}^{D} P(\boldsymbol{\theta}^d|\alpha) \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \\
&= \prod_{d=1}^{D} \frac{\Gamma(\sum_{i=1}^{|\alpha} \alpha_i)}{\prod_{i=1}^{|\alpha|} \Gamma(\alpha_i)} \prod_{k=1}^{L} \theta_{d,k}^{\alpha_k - 1} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)}
\end{aligned}
\tag{F.9}
$$

We now can write the above formulation as:

$$
\begin{aligned}
P(\boldsymbol{Z}|\boldsymbol{\Theta}) \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} &= \prod_{i=1}^{N^d} \theta_{d_i, z_i} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \\
&= \prod_{d=1}^{D} \prod_{k=1}^{L} \theta_{d,k}^{S_{d,k}} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)}
\end{aligned}
\tag{F.10}
$$

S is the count matrix. $S_{d,k}$ is the number of times topic k has been assigned to the a word in the document d. S_d denotes the d^{th} row of S.

$$
\begin{aligned}
P(\boldsymbol{Z}|\alpha) \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} &= \int P(\boldsymbol{Z}|\boldsymbol{\Theta}) P(\boldsymbol{\Theta}|\boldsymbol{\alpha}) \mathrm{d}\boldsymbol{\Theta} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \\
&= \prod_{d=1}^{D} \left(\int \prod_{d=1}^{D} \frac{\Gamma(\sum_{i=1}^{|\alpha} \alpha_i)}{\prod_{i=1}^{|\alpha|} \Gamma(\alpha_i)} \prod_{k=1}^{L} \theta_{d,k}^{S_{d,k}\alpha_k - 1} \mathrm{d}\theta_d \right) \\
&\quad \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \\
&= \prod_{d=1}^{D} \frac{\frac{\prod_{i=1}^{|S_k + \alpha|} \Gamma(S_k + \alpha_i)}{\Gamma(\sum_{i=1}^{|S_k + \alpha|} (S_k + \alpha_i))}}{\frac{\prod_{i=1}^{|\alpha|} \Gamma(\alpha_i)}{\Gamma(\sum_{i=1}^{|\alpha|} \alpha_i)}} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)}
\end{aligned}
\tag{F.11}
$$

Appendix G

Proof of Bayes' Theorem when Cast into an Optimization Problem for Our Bigram Supervised Topic Model

Proof for the optimization equation used in our Bigram Supervised Topic Model

Based on the formula of Bayes' Theorem, we can deduce that $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{B}, \alpha, \beta)$ is the posterior distribution that needs to be found out. $P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{B}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$ is the prior distribution, and the denominator $P(\boldsymbol{B}|\alpha, \beta)$ is the marginal distribution over data.

We know that the Kullback-Leibler Divergence (KLD) from a distribution p to a distribution q can be written as KLD$(q||p)$. Suppose we consider an arbitrary distribution $Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{B}, \alpha, \beta)$. Our goal is to ensure that this distribution is equal to the posterior distribution $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{B}, \alpha, \beta)$. As in the Bayes' rule, this posterior is obtained by iteratively updating the prior $P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{B}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})$.

Suppose we want to minimize the divergence between the arbitrary distribution and the posterior distribution, and this is what we want to achieve so that the two distributions are as close as possible or equal to each other i.e. they overlap. We can write the statement mathematically as:

$$\underset{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}}\,\text{KLD}[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})] \tag{G.1}$$

We know that:

$$P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{B}, \alpha, \beta) = \frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{B}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{B}|\alpha, \beta)} \tag{G.2}$$

So, replacing $P(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\boldsymbol{B}, \alpha, \beta)$ with the right hand side formulation in Equation G.1, we obtain:

$$\underset{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}}\,\text{KLD}\left[Q(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})||\frac{P_0(\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi}|\alpha, \beta)P(\boldsymbol{B}|\boldsymbol{\Theta}, \boldsymbol{Z}, \boldsymbol{\Phi})}{P(\boldsymbol{B}|\alpha, \beta)}\right] \tag{G.3}$$

This equation equivalently can be written as:

$$\mathbb{E}_Q\left[\ln \frac{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})}{\frac{P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\alpha,\beta)P(\boldsymbol{B}|\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})}{P(\boldsymbol{B}|\alpha,\beta)}}\right] \tag{G.4}$$

The above formulation can now be written as:

$$\mathbb{E}_Q\left[\ln \frac{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})}{P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\alpha,\beta)} - \ln P(\boldsymbol{B}|\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}) + \ln P(\boldsymbol{B}|\alpha,\beta)\right] \tag{G.5}$$

This can be further written as:

$$\underset{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}}\,\text{KLD}\left[Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})||\frac{P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\alpha,\beta)P(\boldsymbol{B}|\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})}{P(\boldsymbol{B}|\alpha,\beta)}\right] \tag{G.6}$$

This now simplifies to:

$$\underset{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}}\,\text{KLD}(Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\alpha,\beta)) - \mathbb{E}_{\mathbb{Q}}[\ln P(\boldsymbol{B}|\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})] + \ln P(\boldsymbol{B}|\alpha,\beta) \tag{G.7}$$

We can in-fact drop the last term in Equation G.7 because it does not depend on $\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}$, so we get:

$$\underset{Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})\in\mathbb{P}}{\text{minimize}}\,\text{KLD}(Q(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})||P_0(\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi}|\alpha,\beta)) - \mathbb{E}_Q[\ln P(\boldsymbol{B}|\boldsymbol{\Theta},\boldsymbol{Z},\boldsymbol{\Phi})] \tag{G.8}$$

Appendix H

Bigram Supervised Topic Model - Full Gibbs Sampling Derivation

In order proceed with the derivation of the collapsed Gibbs sampling scheme for our bigram supervised topic model, we first need to define a joint distribution for words and the topics alongwith the regularization effects due to the maximum-margin posterior constraints. This joint distribution is written as:

$$P(\boldsymbol{Z}, \boldsymbol{B}|\alpha, \beta) = P(\boldsymbol{B}|\boldsymbol{Z}, \beta) \times P(\boldsymbol{Z}|\alpha) \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \tag{H.1}$$

Now, we can proceed to expand the above equation:

$$\begin{aligned}
&P(\mathbf{w}, \mathbf{z}|\alpha, \beta) = \int \prod_{d=1}^{D} \prod_{i=1}^{N^d} P(w_i^d|\phi_{z_i^d w_{i-1}^d}) \prod_{z=1}^{L} \left(\prod_{v=1}^{W} P(\phi_{zv}|\beta) \right) \mathrm{d}\Phi \\
&\int \prod_{d=1}^{D} \left(\prod_{i=1}^{N^d} P(z_i^d|\theta^d) P(\theta^d|\alpha) \right) \mathrm{d}\Theta \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \\
&= \int \prod_{z=1}^{L} \prod_{w=1}^{W} \left(\prod_{v=1}^{W} \phi_{zwv}^{m_{zwv}} \frac{\Gamma\Big(\sum_{v=1}^{W} \beta_v\Big)}{\prod_{v=1}^{W} \Gamma(\beta_v)} \prod_{v=1}^{W} \phi_{zwv}^{\beta_v - 1} \right) \mathrm{d}\Phi \times \qquad \text{(H.2)} \\
&\int \prod_{d=1}^{D} \left(\prod_{z=1}^{L} \theta_{dz}^{q_{dz}} \frac{\Gamma\Big(\sum_{z=1}^{L} \alpha_z\Big)}{\prod_{z=1}^{L} \Gamma(\alpha_z)} \prod_{z=1}^{L} \theta_{dz}^{\alpha_z - 1} \right) \mathrm{d}\Theta \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \\
&\propto \underbrace{\prod_{z=1}^{L} \prod_{w=1}^{W} \frac{\prod_{v=1}^{W} \Gamma(m_{zwv} + \beta_v)}{\Gamma(\sum_{v=1}^{W} (m_{zwv} + \beta_v))}}_{\text{Probability of bigram in a topic}} \Bigg(\underbrace{\prod_{d=1}^{D} \frac{\prod_{z=1}^{L} \Gamma(q_{dz} + \alpha_z)}{\Gamma(\sum_{z=1}^{L} (q_{dz} + \alpha_z))}}_{\text{Probability of a topic in a document}} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_{y} (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \overline{\boldsymbol{z}}^d)} \Bigg)
\end{aligned} \tag{H.3}$$

We know that $\Gamma(\alpha) = (\alpha - 1)\Gamma(\alpha - 1)$, so now we can obtain the following conditional probability distribution:

$$P(z_i^d|\mathbf{w}, \mathbf{z}_{\neg i}^d, \alpha, \beta) = \frac{P(w_i^d, z_i^d|\mathbf{w}_{\neg i}^d, \mathbf{z}_{\neg i}^d, \mathbf{z}_{\neg i}^d, \alpha, \beta)}{P(w_i^d|\mathbf{w}_{\neg i}^d, \mathbf{z}_{\neg i}^d, \alpha, \beta)}$$

$$= \left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{L} \left(\alpha_z + q_{dz}\right) - 1} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_y (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \bar{\boldsymbol{z}}^d)} \right) \times \left(\frac{\beta_{w_i^d} + m_{z_i^d w_{i-1}^d w_i^d} - 1}{\sum_{v=1}^{W} \left(\beta_v + m_{z_i^d w_{i-1}^d v}\right) - 1} \right) \tag{H.4}$$

Equivalently, the expression above can be written as:

$$P(z_i^d|\mathbf{w}, \mathbf{z}_{\neg i}^d, \alpha, \beta) = \underbrace{\left(\frac{\alpha_{z_i^d} + q_{dz_i^d} - 1}{\sum_{z=1}^{L} \left(\alpha_z + q_{dz}\right) - 1} \times e^{\boldsymbol{\kappa}^{(*)\intercal} \sum_{d=1}^{D} \sum_y (\lambda_y^d)^* \boldsymbol{\Delta f}(y, \bar{\boldsymbol{z}}^d)} \right)}_{\text{Same as the LDA model}} \times \underbrace{\left(\frac{\beta_{w_i^d} + m_{w_{i-1}^d w_i^d z_i^d} - 1}{\sum_{v=1}^{W} (\beta_v + m_{z_i^d w_{i-1}^d v}) - 1} \right)}_{\text{Bigrams in topics}} \tag{H.5}$$

Bibliography

[1] Brett Adams, Dinh Phung, and Svetha Venkatesh. Discovery of latent subcommunities in a blog's readership. *ACM Transactions on the Web*, 4(3):12:1–12:30, 2010.

[2] Deepak Agnihotri, Kesari Verma, and Priyanka Tripathi. Pattern and cluster mining on text data. In *Proceedings of the 4th International Conference on Communication Systems and Network Technologies*, pages 428–432. IEEE, 2014.

[3] D. Aldous. Exchangeability and related topics. *Ecole d'Ete de Probabilites de Saint-Flour XIII-1983*, pages 1–198, 1985.

[4] James Allan. HARD track overview in TREC 2003 High Accuracy Retrieval from Documents. Technical report, DTIC Document, 2005.

[5] Loulwah Alsumait, Pu Wang, Carlotta Domeniconi, and Daniel Barbar. *Embedding Semantics in LDA Topic Models*, pages 183–204. John Wiley & Sons, Ltd, 2010.

[6] Masaki Aono and Mei Kobayashi. Text document cluster analysis through visualization of 3D projections. In *Data Mining for Service*, pages 271–291. Springer, 2014.

[7] Arthur Asuncion, Padhraic Smyth, Max Welling, David Newman, Ian Porteous, and Scott Triglia. *Distributed Gibbs sampling for latent variable models.* 2012.

[8] Nicola Barbieri, Giuseppe Manco, Ettore Ritacco, Marco Carnuccio, and Antonio Bevacqua. Probabilistic topic models for sequence data. *Machine Learning*, 93(1):5–29, 2013.

[9] N. Bartlett, D. Pfau, and F. Wood. Forgetting counts : Constant memory inference for a dependent Hierarchical Pitman-Yor Process. In *Proceedings of the 26th International Conference on Machine Learning*, pages 63–70, 2010.

[10] Doug Beeferman, Adam Berger, and John Lafferty. Statistical models for text segmentation. *Machine Learning*, 34:177–210, 1999.

[11] J.R. Bellegarda. Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1):76 –84, jan 2000.

[12] Michael Bendersky, W Bruce Croft, and Yanlei Diao. Quality-biased ranking of web documents. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 95–104. ACM, 2011.

[13] Michael W Berry, Susan T Dumais, and Gavin W O'Brien. Using linear algebra for intelligent Information Retrieval. *Society for Industrial and Applied Mathematics Review*, 37(4):573–595, 1995.

[14] Suresh K. Bhavnani. Domain-specific search strategies for the effective retrieval of healthcare and shopping information. In *Human factors in Computing Systems*, pages 610–611, 2002.

[15] David Blei and John Lafferty. Correlated topic models. *Proceedings in the Advances in Neural Information Processing Systems*, 18:147, 2006.

[16] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[17] David M. Blei and Peter I. Frazier. Distance dependent Chinese Restaurant Processes. *The Journal of Machine Learning Research*, 12:2461–2488, November 2011.

[18] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 113–120. ACM, 2006.

[19] David M Blei and John D Lafferty. Topic models. *Text mining: classification, clustering, and applications*, 10:71, 2009.

[20] David M Blei and John D Lafferty. Visualizing topics with multi-word expressions. *arXiv preprint arXiv:0907.1013*, 2009.

[21] David M Blei and Jon D McAuliffe. Supervised topic models. In *Proceedings of the Neural Information Processing Systems*, volume 7, pages 121–128, 2007.

[22] David M Blei and Pedro J Moreno. Topic segmentation with an aspect Hidden Markov Model. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 343–348. ACM, 2001.

[23] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[24] Phil Blunsom, Trevor Cohn, Sharon Goldwater, and Mark Johnson. A note on the implementation of Hierarchical Dirichlet Processes. In *Proceedings of the ACL-International Joint Conference on Natural Language Processing 2009 Conference Short Papers*, ACLShort '09, pages 337–340, Stroudsburg, PA, USA, 2009. Association for Computational Linguistics.

[25] Levent Bolelli, Şeyda Ertekin, and C Lee Giles. Topic and trend detection in text collections using Latent Dirichlet Allocation. In *Proceedings of the European Conference on Information Retrieval*, pages 776–780. Springer, 2009.

[26] Charles Bouveyron and Camille Brunet-Saumard. Model-based clustering of high-dimensional data: a review. *Computational Statistics & Data Analysis*, 71:52–78, 2014.

[27] Jordan Boyd-Graber and David M. Blei. Multilingual topic models for unaligned text. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 75–82, Arlington, Virginia, United States, 2009. AUAI Press.

[28] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[29] Bernard Brosseau-Villeneuve, Jian-Yun Nie, and Noriko Kando. Latent word context model for Information Retrieval. *Information Retrieval*, 17(1):21–51, 2014.

[30] Bertram Bruce, Andee Rubin, and Kathleen S. Starr. Why readability formulas fail? *IEEE Transactions on Professional Communication*, pages 50–52, 1981.

[31] Chris Burges, Tal Shaked, Erin Renshaw, Ari Lazier, Matt Deeds, Nicole Hamilton, and Greg Hullender. Learning to rank using Gradient Descent. In *Proceedings of the 22nd International Conference on Machine Learning*, pages 89–96. ACM, 2005.

[32] Christopher JC Burges, Robert Ragno, and Quoc Viet Le. Learning to rank with nonsmooth cost functions. In *Proceedings of the Neural Information Processing Systems*, volume 6, pages 193–200, 2006.

[33] Róbert Busa-Fekete, Balázs Kégl, Tamás Éltető, and György Szarvas. Tune and mix: learning to rank using ensembles of calibrated multi-class classifiers. *Machine Learning*, 93(2-3):261–292, 2013.

[34] Istvn Br and Jcint Szab. Latent Dirichlet Allocation for automatic document categorization. In Wray Buntine, Marko Grobelnik, Dunja Mladeni, and John Shawe-Taylor, editors, *Machine Learning and Knowledge Discovery in Databases*, volume 5782 of *Lecture Notes in Computer Science*, pages 430–441. Springer Berlin Heidelberg, 2009.

[35] Karla L Caballero, Joel Barajas, and Ram Akella. The generalized Dirichlet distribution in enhanced topic detection. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 773–782. ACM, 2012.

[36] Peng Cai, Wei Gao, Aoying Zhou, and Kam-Fai Wong. Relevant knowledge helps in choosing right teacher: active query selection for ranking adaptation. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124. ACM, 2011.

[37] Kate Cain. Text comprehension and its relation to coherence and cohesion in children's fictional narratives. *British Journal of Developmental Psychology*, 21(3):335–351, 2003.

[38] Juan Cao, Jintao Li, Yongdong Zhang, and Sheng Tang. LDA-based retrieval framework for semantic news video retrieval. In *International Conference on Semantic Computing*, pages 155–160, Sept 2007.

[39] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pages 129–136. ACM, 2007.

[40] George Casella and Edward I. George. Explaining the Gibbs sampler. *The American Statistician*, 46(3):pp. 167–174, 1992.

[41] Ali Taylan Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009, 2009.

[42] Jonathan Chang and David M Blei. Relational topic models for document networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*, pages 81–88, 2009.

[43] Tao-Hsing Chang, Yao-Ting Sung, and Yao-Tung Lee. Evaluating the difficulty of concepts on domain knowledge using latent semantic analysis. In *Proceedings of the International Conference on Asian Language Processing*, pages 193–196. IEEE, 2013.

[44] Berlin Chen. Word topic models for spoken document retrieval and transcription. 8(1):2:1–2:27, March 2009.

[45] Changyou Chen, Lan Du, and Wray Buntine. Sampling table configurations for the Hierarchical Poisson-Dirichlet Process. In *Machine Learning and Knowledge Discovery in Databases*, pages 296–311. Springer, 2011.

[46] Xi Chen and Shihong Chen. Subsequence-based text segmentation and labeling. In *Proceedings of the First International Workshop on Education Technology and Computer Science*, volume 1, pages 582–587. IEEE, 2009.

[47] Jen-Tzung Chien and Chuang-Hua Chueh. Latent Dirichlet language model for speech recognition. In *Spoken Language Technology Workshop*, pages 201–204, Dec 2008.

[48] J.T. Chien and C.H. Chueh. Topic-based hierarchical segmentation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):55–66, 2012.

[49] Gerda Claeskens and Nils Lid Hjort. Model selection and model averaging. *Cambridge Books*, 1993.

[50] Meri Coleman and T. L. Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283–284, 1975.

[51] Kevyn Collins-Thompson, Paul N Bennett, Ryen W White, Sebastian de la Chica, and David Sontag. Personalizing web search results by reading level. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 403–412. ACM, 2011.

[52] Kevyn Collins-Thompson and Jamie Callan. Predicting reading difficulty with statistical language models. *Journal of the Association for Information Science and Technology*, 56(13):1448–1462, November 2005.

[53] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[54] Edgar Dale and Jeanne S. Chall. A formula for predicting readability: Instructions. *Educational Research Bulletin*, 27(2):pp. 37–54, 1948.

[55] Van Dang, Michael Bendersky, and W Bruce Croft. Two-stage learning to rank for Information Retrieval. In *Proceedings of the European Conference in Information Retrieval*, pages 423–434. 2013.

[56] W.M. Darling. *Generalized Probabilistic Topic and Syntax Models for Natural Language Processing*. PhD thesis, 2012.

[57] Paul Deane. A nonparametric method for extraction of candidate phrasal terms. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, ACL '05, pages 605–613, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[58] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990.

[59] Arthur P Dempster, Nan M Laird, Donald B Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[60] P.J. Diggle and R.J. Gratton. Monte Carlo methods of inference for implicit statistical models. *Journal of the Royal Statistical Society*, pages 193–227, 1984.

[61] Chris Ding, Tao Li, and Wei Peng. NMF and PLSI: Equivalence and a hybrid algorithm. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 641–642. ACM, 2006.

[62] Chris Ding, Tao Li, and Wei Peng. On the equivalence between Non-Negative Matrix Factorization and Probabilistic Latent Semantic Indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.

[63] L. Du, W. Buntine, and H. Jin. A segmented topic model based on the two-parameter Poisson-Dirichlet Process. *Machine Learning*, 81(1):5–19, 2010.

[64] William H. Dubay. The principles of readability. *Costa Mesa, CA: Impact Information*, 2004.

[65] Susan T. Dumais. Latent Semantic Indexing (LSI): TREC-3 report. In *Overview of the Third Text REtrieval Conference*, pages 219–230, 1995.

[66] Michael B Eisen, Paul T Spellman, Patrick O Brown, and David Botstein. Cluster analysis and display of Genome-wide expression patterns. *Proceedings of the National Academy of Sciences*, 95(25):14863–14868, 1998.

[67] Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare Voss, and Jiawei Han. Scalable topical phrase mining from text corpora. *Computation and Language - arXiv*, 2014.

[68] Thomas S Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, pages 209–230, 1973.

[69] Evelyn C Ferstl and D Yves von Cramon. The role of coherence and cohesion in text comprehension: an event-related fMRI study. *Cognitive Brain Research*, 11(3):325–340, 2001.

[70] E.B. Fox. *Bayesian Nonparametric Learning of Complex Dynamical Phenomena*. Ph.D. thesis, MIT, Cambridge, MA, 2009.

[71] Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky HDP-HMM with application to speaker diarization. *The Annals of Applied Statistics*, 5(2A):1020–1056, 2011.

[72] Thomas François and Eleni Miltsakaki. Do NLP and machine learning improve traditional readability formulas? In *Proceedings of the 1st Workshop on Predicting and Improving Text Readability for Target Reader Populations*, pages 49–57, Montréal, Canada, June 2012.

[73] Jerome H Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, pages 1189–1232, 2001.

[74] Debasis Ganguly, Johannes Leveling, and GarethJ.F. Jones. Topical relevance model. In Yuexian Hou, Jian-Yun Nie, Le Sun, Bo Wang, and Peng Zhang, editors, *Information Retrieval Technology*, volume 7675 of *Lecture Notes in Computer Science*, pages 326–335. Springer Berlin Heidelberg, 2012.

[75] Jianfeng Gao, Kristina Toutanova, and Wen-tau Yih. Clickthrough-based Latent Semantic Models for web search. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, pages 675–684, New York, NY, USA, 2011. ACM.

[76] Wei Gao and Pei Yang. Democracy is good for ranking: Towards multi-view rank learning and adaptation in web search. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 63–72, New York, NY, USA, 2014. ACM.

[77] Eric Gaussier and Cyril Goutte. Relation between PLSA and NMF and implications. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 601–602. ACM, 2005.

[78] Stuart Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6(6):721–741, 1984.

[79] Kleanthi Georgala, Aris Kosmopoulos, and George Paliouras. Spam filtering: An Active Learning approach using incremental clustering. In *Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics*, page 23. ACM, 2014.

[80] Yogesh Girdhar and Gregory Dudek. Exploring underwater environments with curiosity. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 104–110. IEEE, 2014.

[81] Yogesh Girdhar, Philippe Giguère, and Gregory Dudek. Autonomous adaptive exploration using realtime online spatiotemporal topic modeling. *The International Journal of Robotics Research*, page 0278364913507325, 2013.

[82] Yogesh A. Girdhar, David Whitney, and Gregory Dudek. Curiosity based exploration for learning terrain models. *CoRR*, abs/1310.6767, 2013.

[83] Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 673–680, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[84] Sharon Goldwater, Thomas L Griffiths, and Mark Johnson. A Bayesian framework for word segmentation: Exploring the effects of context. *Cognition*, 112(1):21–54, 2009.

[85] Gene H Golub and Christian Reinsch. Singular Value Decomposition and Least Squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970.

[86] Thomas L Griffiths and Mark Steyvers. Finding scientific topics. *Proceedings of the National academy of Sciences of the United States of America*, 101(Suppl 1):5228–5235, 2004.

[87] Thomas L Griffiths, Mark Steyvers, and Joshua B Tenenbaum. Topics in semantic representation. *Psychological Review*, 114(2):211, 2007.

[88] Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic Markov Models. In *Proceedings of the Artificial Intelligence and Statistics*, pages 163–170, 2007.

[89] Viet Ha-Thuc and Padmini Srinivasan. Topic models and a revisit of text-related applications. In *Proceedings of the 2nd PhD Workshop on Information and Knowledge Management*, PIKM '08, pages 25–32, New York, NY, USA, 2008. ACM.

[90] Viet Ha-Thuc and Padmini Srinivasan. A latent Dirichlet framework for relevance modeling. In GaryGeunbae Lee, Dawei Song, Chin-Yew Lin, Akiko Aizawa, Kazuko Kuriyama, Masaharu Yoshioka, and Tetsuya Sakai, editors, *Information Retrieval Technology*, volume 5839 of *Lecture Notes in Computer Science*, pages 13–25. Springer Berlin Heidelberg, 2009.

[91] Michael A.K. Halliday and Ruqaiya Hasan. *Cohesion in English*. Longman, 1976.

[92] Jiawei Han, Hong Cheng, Dong Xin, and Xifeng Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[93] John A Hartigan. *Clustering algorithms*. John Wiley & Sons, Inc., 1975.

[94] Morgan Harvey, Ian Ruthven, and Mark Carman. Ranking social bookmarks using topic models. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management*, CIKM '10, pages 1401–1404, New York, NY, USA, 2010. ACM.

[95] Eva Hasler, Phil Blunsom, Philipp Koehn, and Barry Haddow. Dynamic topic adaptation for phrase-based MT. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden*, 2014.

[96] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[97] Marti A Hearst. TextTiling: Segmenting text into multi-paragraph subtopic passages. *Computational Linguistics*, 23(1):33–64, 1997.

[98] Michael Heilman, Kevyn Collins-Thompson, and Maxine Eskenazi. An analysis of statistical models and features for reading difficulty prediction. In *Proceedings of the 3rd Workshop on Innovative Use of NLP for Building Educational Applications*, pages 71–79. Association for Computational Linguistics, 2008.

[99] Neil W Henry. Latent structure analysis. *Encyclopedia of Statistical Sciences*, 1983.

[100] Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347, 2013.

[101] Thomas Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 50–57. ACM, 1999.

[102] Thomas Hofmann. Unsupervised learning by Probabilistic Latent Semantic Analysis. *Machine Learning*, 42:177–196, 2001.

[103] Thomas Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, 2004.

[104] Thomas Hofmann and Jan Puzicha. Latent class models for collaborative filtering. In *Proceedings of the International Joint Conferences on Artificial Intelligence*, volume 99, pages 688–693, 1999.

[105] Liangjie Hong and Brian D Davison. Empirical study of topic modeling in Twitter. In *Proceedings of the 1st Workshop on Social Media Analytics*, pages 80–88. ACM, 2010.

[106] Liangjie Hong, Byron Dom, Siva Gurumurthy, and Kostas Tsioutsiouliklis. A time-dependent topic model for multiple text streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 832–840, New York, NY, USA, 2011. ACM.

[107] Liangjie Hong, Dawei Yin, Jian Guo, and Brian D Davison. Tracking trends: incorporating term volume into temporal topic models. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 484–492. ACM, 2011.

[108] Fred M Hoppe. Pólya-like urns and the Ewens' sampling formula. *Journal of Mathematical Biology*, 20(1):91–94, 1984.

[109] Eva Hörster, Rainer Lienhart, and Malcolm Slaney. Image retrieval on large-scale image databases. In *Proceedings of the 6th ACM International Conference on Image and Video Retrieval*, CIVR '07, pages 17–24, New York, NY, USA, 2007. ACM.

[110] Guangyan Huang, Jing He, Yanchun Zhang, Wanlei Zhou, Hai Liu, Peng Zhang, Zhiming Ding, Yue You, and Jian Cao. Mining streams of short text for analysis of world-wide event evolutions. *World Wide Web*, pages 1–17, 2014.

[111] Zahurul Islam, Md Rashedur Rahman, and Alexander Mehler. Readability classification of bangla texts. In *Computational Linguistics and Intelligent Text Processing*, pages 507–518. Springer, 2014.

[112] Tomoharu Iwata and Hiroshi Sawada. Topic model for analyzing purchase data with price information. *Data Mining and Knowledge Discovery*, 26(3):559–573, 2013.

[113] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[114] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[115] Shoaib Jameel and Wai Lam. An n-gram topic model for time-stamped documents. In *Proceedings of the 35th European Conference on Information Retrieval*, pages 292–304. Springer, 2013.

[116] Shoaib Jameel and Wai Lam. A nonparametric n-gram topic model with interpretable latent topics. In *Proceedings of the 9th Asia Information Retrieval Societies Conference*, pages 74–85. Springer, 2013.

[117] Shoaib Jameel and Wai Lam. An unsupervised topic segmentation model incorporating word order. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 203–212. ACM, 2013.

[118] Shoaib Jameel, Wai Lam, Ching-man Au Yeung, and Sheaujiun Chyan. An unsupervised ranking method based on a technical difficulty terrain. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 1989–1992. ACM, 2011.

[119] Shoaib Jameel, Wai Lam, and Xiaojun Qian. Ranking text documents based on conceptual difficulty using term embedding and sequential discourse cohesion. In *Proceedings of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology-Volume 01*, pages 145–152. IEEE Computer Society, 2012.

[120] Shoaib Jameel, Wai Lam, Xiaojun Qian, and Ching-man Au Yeung. An unsupervised technical difficulty ranking model based on conceptual terrain in the latent space. In *Proceedings of the 12th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 351–352. ACM, 2012.

[121] Shoaib Jameel and Xiaojun Qian. An unsupervised technical readability ranking model by building a conceptual terrain in LSI. In *Proceedings of the 8th International Conference on Semantics, Knowledge and Grids*, pages 39–46. IEEE, 2012.

[122] Shoaib Jameel, Xiaojun Qian, and Wai Lam. N-gram fragment sequence based unsupervised domain-specific document readability. In *Proceedings of the 24th International Conference on Computational Linguistics*, pages 1309–1326. ACL, 2012.

[123] Qixia Jiang, Jun Zhu, Maosong Sun, and Eric P Xing. Monte Carlo methods for maximum margin supervised topic models. In *Proceedings of the Neural Information Processing Systems*, pages 1601–1609, 2012.

[124] Ou Jin, Nathan N Liu, Kai Zhao, Yong Yu, and Qiang Yang. Transferring topical knowledge from auxiliary long texts for short text clustering. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 775–784. ACM, 2011.

[125] Yookyung Jo, John E Hopcroft, and Carl Lagoze. The web of topics: discovering the topology of topic evolution in a corpus. In *Proceedings of the 20th International Conference on World Wide Web*, pages 257–266. ACM, 2011.

[126] Thorsten Joachims. Text categorization with Support Vector Machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, volume 1398, pages 137–142. 1998.

[127] Thorsten Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 133–142. ACM, 2002.

[128] Mark Johnson. PCFGs, topic models, adaptor grammars and learning topical collocations and the structure of proper names. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1148–1157, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[129] Nikhil Johri, Dan Roth, and Yuancheng Tu. Experts' retrieval with multiword-enhanced author topic model. In *Proceedings of the The North American Chapter of the Association for Computational Linguistics HLT 2010 Workshop on Semantic Search*, SS '10, pages 10–18, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[130] Rosie Jones, Ravi Kumar, Bo Pang, and Andrew Tomkins. I know what you did last summer: query logs and user privacy. In *Proceedings of the 16th ACM International Conference on Information and Knowledge Management*, pages 909–914. ACM, 2007.

[131] Tapas Kanungo and David Orr. Predicting the readability of short web summaries. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pages 202–211. ACM, 2009.

[132] Rohit J Kate, Xiaoqiang Luo, Siddharth Patwardhan, Martin Franz, Radu Florian, Raymond J Mooney, Salim Roukos, and Chris Welty. Learning to predict readability using diverse linguistic features. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 546–554. Association for Computational Linguistics, 2010.

[133] Noriaki Kawamae. Trend analysis model: trend consists of temporal words, topics, and timestamps. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 317–326. ACM, 2011.

[134] Noriaki Kawamae. Supervised n-gram topic model. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 473–482, New York, NY, USA, 2014. ACM.

[135] Dongwoo Kim and Alice Oh. Accounting for data dependencies within a Hierarchical Dirichlet Process Mixture Model. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, pages 873–878, New York, NY, USA, 2011. ACM.

[136] Hyun Duk Kim, Dae Hoon Park, Yue Lu, and ChengXiang Zhai. Enriching text representation with frequent pattern mining for probabilistic topic modeling. *Journal of American Society for Information Science and Technology*, 49(1):1–10, 2012.

[137] Jin Young Kim, Kevyn Collins-Thompson, Paul N Bennett, and Susan T Dumais. Characterizing web content, user interests, and search behavior by reading level and topic. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 213–222. ACM, 2012.

[138] Jin Young Kim, Kevyn Collins-Thompson, Paul N. Bennett, and Susan T. Dumais. Characterizing web content, user interests, and search behavior by reading level and topic. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 213–222, 2012.

[139] J. Peter Kincaid, Robert P. Fishburne, Richard L. Rogers, and Brad S. Chissom. Derivation of new readability formulas (Automated Readability Index, Fog Count and Flesch Reading Ease formula) for navy enlisted personnel. 1975.

[140] W. Kintsch. The role of knowledge in discourse comprehension: a construction-integration model. *Psychological Review*, 95(2):163, 1988.

[141] Jon Kleinberg. Bursty and hierarchical structure in streams. *Data Mining and Knowledge Discovery*, 7(4):373–397, 2003.

[142] Jon M Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[143] Dan Knights, Michael C Mozer, and Nicolas Nicolov. Detecting topic drift with compound topic models. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2009.

[144] Jol Kuiper and Gert Valkenhoef. Top-level MeSH disease terms are not linearly separable in clinical trial abstracts. In Niels Peek, Roque Marn Morales, and Mor Peleg, editors, *Artificial Intelligence in Medicine*, volume 7885 of *Lecture Notes in Computer Science*, pages 130–134. Springer Berlin Heidelberg, 2013.

[145] Giridhar Kumaran, Rosie Jones, and Omid Madani. Biasing web search results for topic familiarity. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, pages 271–272. ACM, 2005.

[146] Takeshi Kurashima, Tomoharu Iwata, Takahide Hoshide, Noriko Takaya, and Ko Fujimura. Geo topic model: joint modeling of user's activity area and interests for location recommendation. In *Proceedings of the 6th ACM International Conference on Web Search and Data Mining*, WSDM '13, pages 375–384, New York, NY, USA, 2013. ACM.

[147] Simon Lacoste-Julien, Fei Sha, and Michael I Jordan. DiscLDA: Discriminative learning for dimensionality reduction and classification. In *Proceedings of the Neural Information Processing Systems*, volume 83, page 85, 2008.

[148] Hanjiang Lai, Yan Pan, Cong Liu, Liang Lin, and Jie Wu. Sparse learning-to-rank via an efficient primal-dual algorithm. *IEEE Transactions on Computers*, 62(6):1221–1233, 2013.

[149] Balaji Lakshminarayanan and Raviv Raich. Inference in supervised latent Dirichlet allocation. In *IEEE International Workshop on Machine Learning for Signal Processing*, pages 1–6. IEEE, 2011.

[150] Jey Han Lau, Timothy Baldwin, and David Newman. On collocations and topic models. *ACM Transactions on Speech and Language Processing*, 10(3):10:1–10:14, 2013.

[151] Jey Han Lau, Karl Grieser, David Newman, and Timothy Baldwin. Automatic labelling of topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1536–1545, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[152] Daniel D Lee and H Sebastian Seung. Algorithms for Non-Negative Matrix Factorization. In *Proceedings of the Neural Information Processing Systems*, pages 556–562, 2000.

[153] R. Lempel and S. Moran. SALSA: The Stochastic approach for Link-structure Analysis. *ACM Transactions on Information Systems*, 19(2):131–160, April 2001.

[154] Gondy Leroy and James E. Endicott. Combining NLP with evidence-based methods to find text metrics related to perceived and actual text difficulty. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 749–754, 2012.

[155] Gondy Leroy, Trudi Miller, Graciela Rosemblat, and Allen Browne. A balanced approach to health information evaluation: A vocabulary-based Naive Bayes classifier and readability formulas. *Journal of the Association for Information Science and Technology*, 59(9):1409–1419, July 2008.

[156] Dingcheng Li, Swapna Somasundaran, and Amit Chakraborty. A combination of topic models with max-margin learning for relation detection. In *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*, pages 1–9. Association for Computational Linguistics, 2011.

[157] Dingcheng Li, Swapna Somasundaran, and Amit Chakraborty. ERD-MedLDA: Entity relation detection using supervised topic models with maximum margin learning. *Natural Language Engineering*, 18(2):263, 2012.

[158] Hang Li. Learning to rank for Information Retrieval and Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 4(1):1–113, 2011.

[159] Li Li, Chang Liu, Fang Wang, Wei Miao, Jie Zhang, Zhiqian Kang, Yihan Chen, and Luying Peng. Unraveling the hidden heterogeneities of breast cancer based on functional miRNA cluster. *PLOS ONE*, 9(1):e87601, 2014.

[160] Ping Li, Christopher JC Burges, Qiang Wu, JC Platt, D Koller, Y Singer, and S Roweis. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Proceedings of the Neural Information Processing Systems*, volume 7, pages 845–852, 2007.

[161] Wei Li and Andrew McCallum. Pachinko allocation: DAG-structured mixture models of topic correlations. In *Proceedings of the International Conference on Machine Learning*, pages 577–584, 2006.

[162] Wenbo Li, Le Sun, Yuanyong Feng, and Dakun Zhang. Smoothing LDA model for text categorization. In Hang Li, Ting Liu, Wei-Ying Ma, Tetsuya Sakai,

Kam-Fai Wong, and Guodong Zhou, editors, *Information Retrieval Technology*, volume 4993 of *Lecture Notes in Computer Science*, pages 83–94. Springer Berlin Heidelberg, 2008.

[163] Zhisheng Li, Chong Wang, Xing Xie, Xufa Wang, and Wei-Ying Ma. Exploring LDA-based document model for Geographic Information Retrieval. In Carol Peters, Valentin Jijkoun, Thomas Mandl, Henning Mller, DouglasW. Oard, Anselmo Peas, Vivien Petras, and Diana Santos, editors, *Advances in Multilingual and Multimodal Information Retrieval*, volume 5152 of *Lecture Notes in Computer Science*, pages 842–849. Springer Berlin Heidelberg, 2008.

[164] Renjie Liao, Jun Zhu, and Zengchang Qin. Nonparametric Bayesian upstream supervised multi-modal topic models. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, pages 493–502, New York, NY, USA, 2014. ACM.

[165] Xiaojun Lin, Dan Li, and Xihong Wu. A joint topical n-gram language model based on LDA. In *Proceedings of the 2010 2nd International Workshop on Intelligent Systems and Applications*, pages 1–4, 2010.

[166] Robert V Lindsey, William P Headden III, and Michael J Stipicevic. A phrase-discovering topic model using Hierarchical Pitman-Yor Processes. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 214–222. Association for Computational Linguistics, 2012.

[167] Tie-Yan Liu. Learning to rank for Information Retrieval. *Foundations and Trends in Information Retrieval*, 3(3):225–331, 2009.

[168] Xiaoyong Liu, W Bruce Croft, Paul Oh, and David Hart. Automatic recognition of reading levels from user queries. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 548–549. ACM, 2004.

[169] J. Lofberg. YALMIP : a toolbox for modeling and optimization in MATLAB. In *2004 IEEE International Symposium on Computer Aided Control Systems Design*, pages 284 –289, sept. 2004.

[170] Stacy K. Lukins, Nicholas A. Kraft, and Letha H. Etzkorn. Bug localization using Latent Dirichlet Allocation. *Information and Software Technology*, 52(9):972 – 990, 2010.

[171] Dashun Ma, Lan Rao, and Ting Wang. An empirical study of SLDA for Information Retrieval. In MohamedVallMohamed Salem, Khaled Shaalan, Farhad Oroumchian, Azadeh Shakery, and Halim Khelalfa, editors, *Information Retrieval Technology*, volume 7097 of *Lecture Notes in Computer Science*, pages 84–92. Springer Berlin Heidelberg, 2011.

[172] David JC MacKay and Linda C Bauman Peto. A hierarchical Dirichlet language model. *Natural Language Engineering*, 1(3):289–308, 1995.

[173] Craig MacDonald, Rodrygo LT Santos, and Iadh Ounis. The whens and hows of learning to rank for web search. *Information Retrieval*, 16(5):584–628, 2013.

[174] Tomonari Masada, Daiji Fukagawa, Atsuhiro Takasu, Yuichiro Shibata, and Kiyoshi Oguri. Modeling topical trends over continuous time with priors. In *Advances in Neural Networks - ISNN 2010*, pages 302–311. 2010.

[175] Andrew McCallum and Xuerui Wang. A note on topical N-grams. *Department of Computer Science, University of Massachusetts, Amherst*, 2005.

[176] G Harry McLaughlin. SMOG grading: A new readability formula. *Journal of Reading*, 12(8):639–646, 1969.

[177] Danielle S McNamara, Eileen Kintsch, Nancy Butler Songer, and Walter Kintsch. Are good texts always better? interactions of text coherence, background knowledge, and levels of understanding in learning from text. *Cognition and Instruction*, 14(1):1–43, 1996.

[178] Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. Automatic labeling of multinomial topic models. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 490–499, New York, NY, USA, 2007. ACM.

[179] Donald Metzler and W Bruce Croft. A Markov Random Field model for term dependencies. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 472–479. ACM, 2005.

[180] Donald Metzler and W Bruce Croft. Linear feature-based models for Information Retrieval. *Information Retrieval*, 10(3):257–274, 2007.

[181] David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '07, pages 500–509, New York, NY, USA, 2007. ACM.

[182] Tom Minka and Stephen Robertson. Selection bias in the LETOR datasets. In *Special Interest Group on Information Retrieval Workshop on Learning to Rank for Information Retrieval*, pages 48–51, 2008.

[183] Hemant Misra, François Yvon, Joemon M Jose, and Olivier Cappe. Text segmentation via topic modeling: An analytical study. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 1553–1556. ACM, 2009.

[184] Jane Morris and Graeme Hirst. Lexical cohesion computed by thesaural relations as an indicator of the structure of text. *Computational Linguistics*, 17(1):21–48, 1991.

[185] P. Van Mulbregt, I. Carp, L. Gillick, S. Lowe, and J. Yamron. Text segmentation and topic tracking on broadcast news via a Hidden Markov Model approach. In *Proceedings of the 6th International Conference on Spoken Language Processing*, pages 2519–2522, 1998.

[186] Arpita Nagpal, Aman Jatain, and Deepti Gaur. Review based on data clustering algorithms. In *Proceedings of the IEEE Conference on Information & Communication Technologies*, pages 298–303. IEEE, 2013.

[187] Makoto Nakatani, Adam Jatowt, and Katsumi Tanaka. Easiest-first search: towards comprehension-based web search. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, pages 2057–2060. ACM, 2009.

[188] Makoto Nakatani, Adam Jatowt, and Katsumi Tanaka. Adaptive ranking of search results by considering user's comprehension. In *Proceedings of the 4th International Conference on Uniquitous Information Management and Communication*, pages 27:1–27:10, 2010.

[189] Ramesh Nallapati. Discriminative models for Information Retrieval. In *Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 64–71. ACM, 2004.

[190] R.M. Neal. Markov chain sampling methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics*, 9(2):249–265, 2000.

[191] Duc Thien Nguyen, William Yeoh, and Hoong Chuin Lau. Distributed Gibbs: A memory-bounded sampling-based DCOP algorithm. In *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, pages 167–174. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[192] Viet-An Nguyen, Jordan Boyd-Graber, and Philip Resnik. SITS: a hierarchical nonparametric model using speaker identity for topic segmentation in multiparty conversations. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers - Volume 1*, ACL '12, pages 78–87, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[193] Uri Nodelman, Christian R Shelton, and Daphne Koller. Continuous time Bayesian networks. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 378–387. Morgan Kaufmann Publishers Inc., 2002.

[194] Hiroshi Noji, Daichi Mochihashi, and Yusuke Miyao. Improvements to the Bayesian topic n-gram models. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing*, pages 1180–1190, 2013.

[195] Peter Orbanz. *Infinite-Dimensional Exponential Families in Cluster Analysis of Structured Data*. PhD thesis, ETH Zurich, 2008.

[196] Adler J Perotte, Frank Wood, Noémie Elhadad, and Nicholas Bartlett. Hierarchically supervised Latent Dirichlet Allocation. In *Proceedings of the Neural Information Processing Systems*, pages 2609–2617, 2011.

[197] Sarah E. Petersen and Mari Ostendorf. A machine learning approach to reading level assessment. *Computer Speech and Language*, 23(1):89–106, January 2009.

[198] Saa Petrovi, Jan najder, and Bojana Dalbelo Bai. Extending lexical association measures for collocation extraction. *Computer Speech and Language*, 24(2):383 – 394, 2010.

[199] Lev Pevzner and Marti A. Hearst. A critique and improvement of an evaluation metric for text segmentation. *Computational Linguistics*, 28(1):19–36, 2002.

[200] Xuan-Hieu Phan, Le-Minh Nguyen, and Susumu Horiguchi. Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, pages 91–100, New York, NY, USA, 2008. ACM.

[201] Do Viet Phuong and Tu Minh Phuong. A keyword-topic model for contextual advertising. In *Proceedings of the 3rd Symposium on Information and Communication Technology*, SoICT '12, pages 63–70, New York, NY, USA, 2012. ACM.

[202] Emily Pitler and Ani Nenkova. Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 186–195. Association for Computational Linguistics, 2008.

[203] J. Pitman and M. Yor. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.

[204] Jim Pitman. *Combinatorial stochastic processes*, volume 1875. Springer-Verlag, 2006.

[205] Tamara Polajnar and Mark Girolami. Application of lexical topic models to protein interaction sentence prediction. 2009.

[206] Mark Polczynski and Michael Polczynski. Using the k-Means clustering algorithm to classify features for Choropleth Maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 49(1):69–75, 2014.

[207] Alexandrin Popescul, David M. Pennock, and Steve Lawrence. Probabilistic models for unified collaborative and content-based recommendation in sparse-data environments. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, UAI'01, pages 437–444, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[208] Ian Porteous, David Newman, Alexander Ihler, Arthur Asuncion, Padhraic Smyth, and Max Welling. Fast collapsed Gibbs sampling for Latent Dirichlet Allocation. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 569–577, New York, NY, USA, 2008. ACM.

[209] I. Pruteanu-Malinici, Lu Ren, J. Paisley, E. Wang, and L. Carin. Hierarchical Bayesian modeling of topics in time-stamped documents. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):996–1011, 2010.

[210] Tao Qin, Tie-Yan Liu, Jun Xu, and Hang Li. LETOR: A benchmark collection for research on learning to rank for Information Retrieval. *Information Retrieval*, 13(4):346–374, 2010.

[211] Zengchang Qin, Marcus Thint, and Zhiheng Huang. Ranking answers by hierarchical topic models. In Been-Chian Chien, Tzung-Pei Hong, Shyi-Ming Chen, and Moonis Ali, editors, *Next-Generation Applied Intelligence*, volume 5579 of *Lecture Notes in Computer Science*, pages 103–112. Springer Berlin Heidelberg, 2009.

[212] Rani Qumsiyeh and Yiu-Kai Ng. ReadAid: A robust and fully-automated readability assessment tool. In *Proceedings of the 23rd IEEE International Conference on Tools with Artificial Intelligence*, pages 539–546, 2011.

[213] Dheeraj Rajagopal, Daniel Olsher, Erik Cambria, and Kenneth Kwok. Commonsense-based topic modeling. In *Proceedings of the 2nd International Workshop on Issues of Sentiment Discovery and Opinion Mining*, WISDOM '13, pages 6:1–6:8, New York, NY, USA, 2013. ACM.

[214] Daniel Ramage, David Hall, Ramesh Nallapati, and Christopher D Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 248–256. Association for Computational Linguistics, 2009.

[215] Santu Rana, Dinh Phung, and Svetha Venkatesh. Split-merge augmented Gibbs sampling for Hierarchical Dirichlet Processes. In Jian Pei, VincentS. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*, volume 7819 of *Lecture Notes in Computer Science*, pages 546–557. Springer Berlin Heidelberg, 2013.

[216] Joseph Reisinger, Austin Waters, Bryan Silverthorn, and Raymond J Mooney. Spherical topic models. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 903–910, 2010.

[217] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[218] Jeffrey C Reynar. *Topic segmentation: Algorithms and applications*. PhD thesis, University of Pennsylvania, 1998.

[219] Martin Riedl and Chris Biemann. How text segmentation algorithms gain from topic models? In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 553–557. Association for Computational Linguistics, 2012.

[220] Martin Riedl and Chris Biemann. Topictiling: a text segmentation algorithm based on LDA. In *Proceedings of the Association for Computational Linguistics 2012 Student Research Workshop*, pages 37–42. Association for Computational Linguistics, 2012.

[221] S.E. Robertson, S. Walker, S. Jones, M.M. Hancock-Beaulieu, and M. Gatford. Okapi at TREC-3. pages 109–126, 1996.

[222] Michal Rosen-Zvi, Thomas Griffiths, Mark Steyvers, and Padhraic Smyth. The author-topic model for authors and documents. In *Proceedings of the 20th Conference on Uncertainty in Artificial Intelligence*, pages 487–494. AUAI Press, 2004.

[223] Timothy N Rubin, America Chambers, Padhraic Smyth, and Mark Steyvers. Statistical topic models for multi-label document classification. *Machine Learning*, 88(1-2):157–208, 2012.

[224] Andres Salcedo, AA Berlind, and A Maller. Dark matter halo clustering in the lasdamas simulations. In *American Astronomical Society Meeting Abstracts*, volume 223, 2014.

[225] G. Salton, A. Wong, and C. S. Yang. A Vector Space Model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[226] Kim Schouten and Flavius Frasincar. A dependency graph isomorphism for news sentence searching. In Elisabeth Mtais, Farid Meziane, Mohamad Saraee, Vijayan Sugumaran, and Sunil Vadera, editors, *Natural Language Processing and Information Systems*, volume 7934 of *Lecture Notes in Computer Science*, pages 384–387. Springer Berlin Heidelberg, 2013.

[227] Sarah E Schwarm and Mari Ostendorf. Reading level assessment using Support Vector Machines and statistical language models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 523–530. Association for Computational Linguistics, 2005.

[228] R.J. Senter and E.A. Smith. Automated Readability Index. *Cincinnati University Ohio*, 1967.

[229] J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

[230] M. Shafiei and Evangelos Milios. A statistical model for topic segmentation and clustering. In *Proceedings of the Advances in Artificial Intelligence*, volume 5032, pages 283–295, 2008.

[231] M Mahdi Shafiei and Evangelos E Milios. Latent Dirichlet co-clustering. In *Proceedings of the 6th International Conference on Data Mining*, pages 542–551. IEEE, 2006.

[232] PA Shaver. The clustering of Quasars. *Astronomy and Astrophysics*, 136:L9, 1984.

[233] Jianfeng Si, Qing Li, Tieyun Qian, and Xiaotie Deng. Users interest grouping from online reviews based on topic frequency and order. *World Wide Web*, pages 1–22, 2013.

[234] Luo Si and Jamie Callan. A statistical model for scientific readability. In *Proceedings of the 10th International Conference on Information and Knowledge Management*, pages 574–576. ACM, 2001.

[235] Craig Silverstein, Hannes Marais, Monika Henzinger, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1):6–12, September 1999.

[236] Alessandro Sordoni, Jing He, and Jian-Yun Nie. Modeling latent topic interactions using quantum interference for Information Retrieval. In *Proceedings of the 22nd ACM International Conference on Conference on Information & Knowledge Management*, pages 1197–1200. ACM, 2013.

[237] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of Latent Semantic Analysis*, 427(7):424–440, 2007.

[238] Amos J. Storkey and Andrew Dai. The supervised hierarchical Dirichlet Process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99, 2014.

[239] Erik Blaine Sudderth. *Graphical models for visual object recognition and tracking.* PhD thesis, Massachusetts Institute of Technology, 2006.

[240] Russell Swan and James Allan. Extracting significant time varying features from text. In *Proceedings of the 8th International Conference on Information and Knowledge Management*, pages 38–45. ACM, 1999.

[241] Andrea Tagarelli and George Karypis. A segment-based approach to clustering multi-topic documents. *Knowledge and Information Systems*, 34(3):563–595, 2013.

[242] Chenhao Tan, Evgeniy Gabrilovich, and Bo Pang. To each his own: personalized content selection based on text comprehensibility. In *Proceedings of the 5th ACM International Conference on Web Search and Data Mining*, pages 233–242. ACM, 2012.

[243] Jian Tang, Ning Liu, Jun Yan, Yelong Shen, Shaodan Guo, Bin Gao, Shuicheng Yan, and Ming Zhang. Learning to rank audience for behavioral targeting in display ads. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 605–610. ACM, 2011.

[244] Yee Whye Teh. A hierarchical Bayesian language model based on Pitman-Yor Processes. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 985–992, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[245] Yee Whye Teh. Dirichlet Process, 2010.

[246] Yee Whye Teh and Michael I Jordan. Hierarchical Bayesian nonparametric models with applications. *Bayesian Nonparametrics: Principles and Practice*, pages 158–207, 2010.

[247] Yee Whye Teh, Michael I Jordan, Matthew J Beal, and David M Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

[248] Yee Whye Teh, David Newman, and Max Welling. A collapsed variational Bayesian inference algorithm for Latent Dirichlet Allocation. In *Proceedings of the Neural Information Processing Systems*, volume 6, pages 1378–1385, 2006.

[249] Philip Van Oosten and Véronique Hoste. Readability annotation: Replacing the expert by the crowd. In *Proceedings of the 6th Workshop on Innovative Use of NLP for Building Educational Applications*, pages 120–129, 2011.

[250] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.

[251] Ivan Vulic and Marie-Francine Moens. A unified framework for monolingual and cross-lingual relevance modeling based on probabilistic topic models. In *Proceedings of the 35th European Conference on Information Retrieval*, volume 7814 of *Lecture Notes in Computer Science*, pages 98–109. Springer Berlin Heidelberg, 2013.

[252] Hanna M Wallach. Topic modeling: beyond bag-of-words. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 977–984. ACM, 2006.

[253] Hanna M Wallach, Iain Murray, Ruslan Salakhutdinov, and David Mimno. Evaluation methods for topic models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1105–1112. ACM, 2009.

[254] Ai Wang, YaoDong Li, and Wei Wang. Cross language information retrieval based on LDA. In *IEEE International Conference on Intelligent Computing and Intelligent Systems*, volume 3, pages 485–490, Nov 2009.

[255] Chi Wang, Marina Danilevsky, Nihit Desai, Yinan Zhang, Phuong Nguyen, Thrivikrama Taula, and Jiawei Han. A phrase mining framework for recursive construction of a topical hierarchy. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pages 437–445, New York, NY, USA, 2013. ACM.

[256] Chong Wang, David Blei, and David Heckerman. Continuous time dynamic topic models. In *Proceedings of the Uncertainty in Artificial Intelligence*, pages 579–586, 2008.

[257] Chong Wang, David Blei, and Fei-Fei Li. Simultaneous image classification and annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1903–1910. IEEE, 2009.

[258] Chong Wang and David M. Blei. A split-merge MCMC algorithm for the Hierarchical Dirichlet Process. *Computing Research Repository*, abs/1201.1657, 2012.

[259] Sheng Wang, Fangtao Li, and Ming Zhang. Supervised topic model with consideration of user and item. In *Workshops at the 27th AAAI Conference on Artificial Intelligence*, 2013.

[260] Xuerui Wang and Andrew McCallum. Topics over time: a non-Markov continuous-time model of topical trends. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 424–433. ACM, 2006.

[261] Xuerui Wang, Andrew McCallum, and Xing Wei. Topical n-grams: Phrase and topic discovery, with an application to Information Retrieval. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 697–702. IEEE, 2007.

[262] Xuerui Wang, Natasha Mohanty, and Andrew McCallum. Group and topic discovery from relations and their attributes. *Proceedings of the Neural Information Processing Systems*, 18:1449, 2006.

[263] Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and Edward Y Chang. PLDA: Parallel Latent Dirichlet Allocation for large-scale applications. In *Algorithmic Aspects in Information and Management*, pages 301–314. Springer, 2009.

[264] Yi Wang, Hongjie Bai, Matt Stanton, Wen-Yen Chen, and EdwardY. Chang. PLDA: Parallel latent Dirichlet allocation for large-scale applications. In Andrew V. Goldberg and Yunhong Zhou, editors, *Algorithmic Aspects in Information and Management*, volume 5564 of *Lecture Notes in Computer Science*, pages 301–314. Springer Berlin Heidelberg, 2009.

[265] Yu Wang, Eugene Agichtein, and Michele Benzi. TM-LDA: efficient online modeling of latent topic transitions in social media. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 123–131. ACM, 2012.

[266] Xing Wei and W Bruce Croft. LDA-based document models for ad-hoc retrieval. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 178–185. ACM, 2006.

[267] Xing Wei and W. Bruce Croft. Investigating retrieval performance with manually-built topic models. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, RIAO '07, pages 333–349, Paris, France, France, 2007.

[268] Jason Weston, Hector Yee, and Ron J Weiss. Learning to rank recommendations with the k-order statistic loss. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 245–248. ACM, 2013.

[269] Ryen W White, Susan T Dumais, and Jaime Teevan. Characterizing the influence of domain expertise on web search behavior. In *Proceedings of the 2nd ACM International Conference on Web Search and Data Mining*, pages 132–141. ACM, 2009.

[270] Qiang Wu, Christopher JC Burges, Krysta M Svore, and Jianfeng Gao. Adapting boosting for Information Retrieval measures. *Information Retrieval*, 13(3):254–270, 2010.

[271] Wensheng Wu and Tingting Zhong. Searching the deep web using proactive phrase queries. In *Proceedings of the 22nd International Conference on World Wide Web Companion*, pages 137–138. International World Wide Web Conferences Steering Committee, 2013.

[272] Han Xiao and Thomas Stibor. Efficient collapsed gibbs sampling for Latent Dirichlet Allocation. *Journal of Machine Learning Research-Proceedings Track*, 13:63–78, 2010.

[273] Boyi Xie and Rebecca J Passonneau. Supervised HDP using prior knowledge. In *Natural Language Processing and Information Systems*, pages 197–202. Springer, 2012.

[274] Jun Xu and Hang Li. AdaRank: A boosting algorithm for Information Retrieval. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 391–398. ACM, 2007.

[275] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on Non-Negative Matrix Factorization. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval*, pages 267–273. ACM, 2003.

[276] Wael MS Yafooz, Siti ZZ Abidin, Nasiroh Omar, and Rosenah A Halim. Shared-table for textual data clustering in distributed relational databases. In *Proceedings of the 1st International Conference on Advanced Data and Information Engineering*, pages 49–57. Springer, 2014.

[277] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 1445–1456. International World Wide Web Conferences Steering Committee, 2013.

[278] Xin Yan, Raymond YK Lau, Dawei Song, Xue Li, and Jian Ma. Toward a semantic granularity model for domain-specific Information Retrieval. *ACM Transactions on Information Systems*, 29(3):15, 2011.

[279] Xin Yan, Dawei Song, and Xue Li. Concept-based document readability in domain specific Information Retrieval. In *Proceedings of the 15th ACM International Conference on Information and Knowledge Management*, pages 540–549. ACM, 2006.

[280] Zehua Yan and Fang Li. News thread extraction based on topical n-gram model with a background distribution. In Bao-Liang Lu, Liqing Zhang, and James Kwok, editors, *Proceedings of the Neural Information Processing*, volume 7063 of *Lecture Notes in Computer Science*, pages 416–424. Springer Berlin Heidelberg, 2011.

[281] Chyi-Kwei Yau, Alan Porter, Nils Newman, and Arho Suominen. Clustering scientific documents with topic modeling. *Scientometrics*, pages 1–20, 2014.

[282] Xing Yi and James Allan. Evaluating topic models for Information Retrieval. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, CIKM '08, pages 1431–1432, New York, NY, USA, 2008. ACM.

[283] Zhijun Yin, Liangliang Cao, Jiawei Han, Cheng Xiang Zhai, and Thomas Huang. LPTA: A probabilistic model for latent periodic topic analysis. In *Proceedings of the IEEE 11th International Conference on Data Mining*, pages 904–913. IEEE, 2011.

[284] Kazuyoshi Yoshii and Masataka Goto. A vocabulary-free infinity-gram model for nonparametric Bayesian chord progression analysis. In *Proceedings of the International Society for Music Information Retrieval*, pages 645–650, 2011.

[285] Dingrong Yuan, Yuwei Cuan, and Yaqiong Liu. An effective clustering algorithm for transaction databases based on k-Mean. *Journal of Computers*, 9(4):812–816, 2014.

[286] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A Support Vector method for optimizing average precision. In *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 271–278. ACM, 2007.

[287] Arnold Zellner. Optimal information processing and Bayes's theorem. *The American Statistician*, 42(4):278–280, 1988.

[288] Chengxiang Zhai and John Lafferty. A study of smoothing methods for language models applied to Information Retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, April 2004.

[289] Lidon Zhai, Zhaoyun Ding, Yan Jia, and Bin Zhou. A word position-related LDA model. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(06):909–925, 2011.

[290] Ce Zhang and Christopher Ré. Towards high-throughput Gibbs sampling at scale: A study across storage managers. In *Proceedings of the 2013 International Conference on Management of Data*, pages 397–408. ACM, 2013.

[291] Chenyi Zhang and Jianling Sun. Large scale microblog mining using distributed MB-LDA. In *Proceedings of the 21st International Conference Companion on World Wide Web*, WWW '12 Companion, pages 1035–1042, New York, NY, USA, 2012. ACM.

[292] Xiangmin Zhang, Jingjing Liu, Michael Cole, and Nicholas Belkin. Predicting users' domain knowledge in Information Retrieval using multiple regression analysis of search behaviors. *Journal of the Association for Information Science and Technology*, 2014.

[293] Jin Zhao and Min-Yen Kan. Domain-specific iterative readability computation. In *Proceedings of the 10th Annual Joint Conference on Digital Libraries*, pages 205–214. ACM, 2010.

[294] Shibin Zhou, Kan Li, and Yushu Liu. Text categorization based on topic model. In Guoyin Wang, Tianrui Li, JerzyW. Grzymala-Busse, Duoqian Miao, Andrzej Skowron, and Yiyu Yao, editors, *Rough Sets and Knowledge Technology*, volume 5009 of *Lecture Notes in Computer Science*, pages 572–579. Springer Berlin Heidelberg, 2008.

[295] Shibin Zhou, Kan Li, and Yushu Liu. Text categorization based on topic model. *International Journal of Computational Intelligence Systems*, 2(4):398–409, 2009.

[296] TomChao Zhou, MichaelRung-Tsong Lyu, Irwin King, and Jie Lou. Learning to suggest questions in social media. *Knowledge and Information Systems*, pages 1–28, 2014.

[297] Jun Zhu, Amr Ahmed, and Eric P Xing. MedLDA: maximum margin supervised topic models for regression and classification. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1257–1264. ACM, 2009.

[298] Jun Zhu, Amr Ahmed, and Eric P Xing. MedLDA: maximum margin supervised topic models. *The Journal of Machine Learning Research*, 13(1):2237–2278, 2012.

[299] Jun Zhu, Ning Chen, Hugh Perkins, and Bo Zhang. Gibbs max-margin topic models with fast sampling algorithms. In *Proceedings of the 30th International Conference on Machine Learning*, pages 124–132, 2013.

[300] Jun Zhu, Xun Zheng, and Bo Zhang. Improved Bayesian logistic supervised topic models with data augmentation. In *Proceedings of the Association for Computational Linguistics*, pages 187–195, 2013.

[301] Jun Zhu, Xun Zheng, Li Zhou, and Bo Zhang. Scalable inference in max-margin topic models. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 964–972. ACM, 2013.

The End

If we knew what it was we were doing, it would not be called research, would it? - Albert Einstein