Ranking Text Documents Based on Conceptual Difficulty using Term Embedding and Sequential Discourse Cohesion

Shoaib Jameel , Wai Lam , Xiaojun Qian
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong
{*msjameel, wlam, xjqian*}*@se.cuhk.edu.hk*

***Abstract*—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. Our approach also considers transitions between the segments generated in sequence. It performs better than the current state-of-the-art comparative methods.**

***Keywords*-Conceptual Difficulty; LSI; Term Embedding; K-means**

I. Introduction

It has been studied that an increasing number of people often search for information outside their domain of specialization [1]. But it becomes difficult to automatically determine the expertise level of the user from the query due to the fact that many queries are short and ambiguous [2] and may not directly indicate the true reading expertise of the searcher. Recently, Google has tried to address the problem by introducing a new interface under "More Search Tools" to let users specify their reading level. We investigate a novel unsupervised framework for determining the conceptual difficulty of a text document. Our model can be readily applied in domain-specific vertical search engines to re-rank documents according to their conceptual difficulty.

Our model makes use of Latent Semantic Indexing (LSI) [3] to migrate from the word space to the concept space. We compute each term's conceptual difficulty based on its geometry in the latent space. Our model also considers the notion of "conceptual transitions" in the concept space. The value aggregated after sequential term scanning for a document quantifies the conceptual difficulty of that document. This value will be used to re-rank the results obtained from a general purpose IR system.

The work described in this paper is substantially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: CUHK413510). This work is also affiliated with the Microsoft-CUHK Joint Laboratory for Human-centric Computing and Interface Technologies.

A note on the use of terminology. Our work mainly falls within the purview of determining the reading difficulty of text documents. The term readability has been used in different ways [4] in different works such as concept based readability in [5], comprehensibility in [4], domain-specific iterative readability in [6] and technical difficulty in [7] and [8]. We focus on domain-specific documents, and hence use the term conceptual difficulty in this paper. Conceptual difficulty relates to finding the reading difficulty of a domain-specific document.

Our main contribution mainly lies in the way we digress from traditional readability based approaches and use a conceptual model to score the technical importance of a term based on its co-occurrence. We propose a linear embedding model to linearly embed a term vector with the document vectors in the concept space. We measure cohesion based on similarity between the segments using an unsupervised approach. We conduct extensive experiments and show the effectiveness of our approach.

II. Related Work

The problem of determining the reading difficulty in domain-specific IR is not new. For example, in [5] the authors presented concept based readability method where they used a domain-specific ontology to compute document scope and document cohesion of domain-specific terms in a document. One limitation of their approach is the requirement of a domain-specific ontology to capture domain-specific terms in a document. Zhao et al., [6] tried to address this problem where they proposed a domain-specific iterative readability computation method. Their approach is based on the notion of popular link analysis algorithms such as HITS and SALSA. However, their method relies on some seed set of domain-specific terms. This is problematic as one would need seed set of terms for every domain. We addressed the problem in our preliminary work where we presented technical difficulty models in [7] and [8]. We used Latent Semantic Indexing (LSI) [9] to capture domain-specific terms in a document. Although we could achieve

978-0-7695-4880-7/12 $26.00 © 2012 IEEE
DOI 10.1109/WI-IAT.2012.235

better results than the traditional readability methods, our preliminary approach lacked a solid theoretical foundation.

Traditional readability methods [10] have been used in several applications such as finding grade level of texts. These methods mainly consider surface level features of texts such as number of syllables, sentence length, word length etc. They also fail to give precise prediction on web pages [11]. One shortcoming is that readability methods completely disregard an understanding of the complexity of ideas inherent in text [12]. These methods simply look at the surface-level features. Another shortcoming is that readability methods do not consider cohesion, which forms one of the ingredients in comprehension and computing the difficulty of texts [13]. In addition, they completely disregard the context in which a term has been used [7]. For example, "star" comprising of one syllable can be interpreted as a domain-specific term in Science and Astronomy. In contrast, it can be treated as a common term in other domains, for example, Movies. Despite having several shortcomings, they still remain a dominant tool for finding the reading difficulty of texts because of their ease of use and simplicity.

Some supervised learning approaches have also been adopted to tackle the problem of readability. In [11], the authors discussed smoothed unigram model to predict readability of texts. The authors have used a small corpus of text documents classified into different American grade levels and built a classifier based on a unigram language model to predict readability of texts. Bendersky et al., [14] described a quality biased approach to improve ranking in a general purpose web search engine where the authors used readability and cohesive nature of texts as one of the features in ranking. In [15], the authors used SVM to predict reading difficulty of texts using syntactic and vocabulary based features. Kumaran et al., [16] described topic familiarity problem in texts. They have mentioned that topic familiarity is different from traditional readability. They have used several readability features in order to train a classifier to predict reading difficulty of texts. The authors found that stopwords is a useful feature in their classifier. Heilman et al., [17] used grammatical features to train their classifier. Pitler et al., [18] described several linguistic features in their classifier and obtained significant results. In [19] the authors have used diverse features from text including language model to train a classifier. Determination of reading level from queries has been described in [20] where the authors trained a support vector machine. Recently, few works have tried to address readability problem by building user models and personalizing search results [4], [21] and [22]. This direction requires query log data with individual user session details. This could lead to privacy problems [23]. One limitation in supervised learning approaches is that it requires annotated data to train a classifier. Obtaining annotated data might be expensive and time consuming at times.

When elements of text tend to hang together [24], the state is called cohesion. Cohesion helps in comprehension [13]. Halliday and Hasan [13] stated that the start of the text will not be cohesive with the later sections of text. We use this conclusion in this paper where we consider that maintaining the term order in a text document is important.

III. Sequential Term Transition Model (STTM)

A. 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. We group multiple terms in sequence into variable length segments and measure similarity between the sequences of segments in a document.

B. The Latent Space

We make use of Latent Semantic Indexing (LSI) to derive latent information that plays a major role in our framework. One essential component in LSI is Singular Value Decomposition (SVD). Consider a domain, the input to LSI is a $T \times D$ term-document matrix, $\mathbf{W}$, where T 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 denoted as (idf). SVD factorizes $\mathbf{W}$ into three matrices as shown in Equation 1.

$$\mathbf{W} \approx \hat{\mathbf{W}} = \mathbf{U}\mathbf{S}\mathbf{V}^T \tag{1}$$

where $\mathbf{U}$ is a $T \times f$ 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 $f << \min(T, D)$ is the number of factors, and $\mathbf{V}^T$ denotes matrix transposition of $\mathbf{V}$.

Traditional vector space model [25] cannot find new structural relationships between terms and their documents in the collection [26]. 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 [7] and [8]. 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.

C. 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 [5] 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 1. Suppose that $\mathbf{U}$ and $\mathbf{S}$ are matrices in SVD computation as expressed in Equation 1. Let $\mathbf{R}$ be a matrix with dimension $T \times f$ and $\mathbf{R}$ is computed by matrix multiplication of $\mathbf{U}$ and $\mathbf{S}$ as depicted in Equation 2.

$$\mathbf{R} = \mathbf{U} \times \mathbf{S} \tag{2}$$

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 3.

$$\mathbf{R} = \begin{bmatrix} \vec{r}_1 \\ \vec{r}_2 \\ \cdots \\ \vec{r}_x \\ \cdots \\ \vec{r}_T \end{bmatrix} \tag{3}$$

We normalize $\vec{r}_x$ as follows:

$$\hat{\vec{r}}_x = \frac{\vec{r}_x}{||\vec{r}_x||} \tag{4}$$

Let $\mathbf{L}$ be a matrix of dimension $f \times D$ and $\mathbf{L}$ is computed by a matrix multiplication of $\mathbf{S}$ and $\mathbf{V}^T$ as depicted in Equation 5

$$\mathbf{L} = \mathbf{S} \times \mathbf{V}^T \tag{5}$$

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 6.

$$\mathbf{L} = \begin{bmatrix} \vec{l}_1 \\ \vec{l}_2 \\ \cdots \\ \vec{l}_j \\ \cdots \\ \vec{l}_D \end{bmatrix}^T \tag{6}$$

We normalize each document vector $\vec{l}_j$ as depicted in Equation 7.

$$\hat{\vec{l}}_j = \frac{\vec{l}_j}{||\vec{l}_j||} \tag{7}$$

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 $\{q_1, q_2, \cdots, q_{N_x}\}$ where N_x 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 $\hat{\vec{l}}_{q_i}$. The dimension of $\hat{\mathbf{L}}_x$ is $N_x \times f$. As a result, $\hat{\mathbf{L}}_x$ can be expressed as depicted in Equation 8

$$\hat{\mathbf{L}}_x = \begin{bmatrix} \vec{l}_{q_1} \\ \vec{l}_{q_2} \\ \cdots \\ \vec{l}_{q_{N_x}} \end{bmatrix} \tag{8}$$

The term linear embedding problem can be formulated as minimizing the distance expressed in Equation 9.

$$\begin{aligned} \underset{[\gamma_n^x]}{\text{minimize}} \quad & ||\hat{\vec{r}}_x - [\gamma_n^x]^T \hat{\mathbf{L}}_x|| \\ \text{subject to} \quad & \sum_{n=1}^{N_x} \gamma_n^x = 1, \gamma_n^x \geq 0 \end{aligned} \tag{9}$$

The weights encapsulated in $[\gamma_n^x]$ by linear synthesis in Equation 9 can be regarded as technical contribution that the term plays in the document. The dimension of $[\gamma_n^x]$ is $N_x \times 1$. By adopting the optimization in Equation 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 $\hat{\vec{r}}_x$. 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 for each term in the vocabulary. Consider document j from the entire collection. Let C_j 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:

$$\chi_j = \frac{\sum_{i=1}^{C_j} \gamma_j^{t_i}}{C_j} \tag{10}$$

D. Sequential Segment Cohesion

As [13] 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 [27]. 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.

Algorithm 1: Cohesion based on segmentation.

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$\big(\frac{\zeta_j}{S_j} \times \tau\big)$;

Generally the latent space obtained via SVD does not directly provide a reasonable cluster membership of every term in space [28]. A clustering algorithm is needed. In [26], 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 [29]. We cluster low-dimensional term vectors in the latent space. The input to the clustering algorithm are the normalized low-dimensional term vectors $\hat{\vec{r}}_x$ as depicted in Equation 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 1, 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:

$$\zeta_j = \frac{\sum_{s=1}^{S_j-1} \nu(\vec{\Delta}_s, \vec{\Delta}_{s+1})}{S_j} \tau \tag{11}$$

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 pseudocode in Algorithm 1. 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.

IV. 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

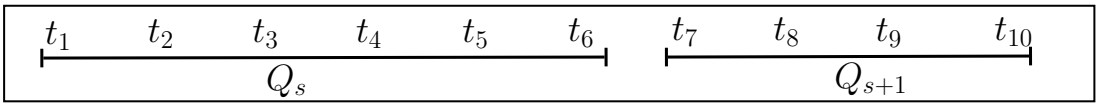

Figure 1. One particular term sequence $(t_1, t_2, \cdots, t_{10})$ with two segments (Q_s, Q_{s+1}) in sequence.

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 I
CONCEPTUAL DIFFICULTY JUDGMENT GUIDELINES GIVEN TO THE HUMAN JUDGES.

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 discourse [30]. Therefore, conceptual difficulty, Φ_j of a document can be formulated as:

$$\Phi_j = \beta\chi_j + (1-\beta)\frac{1}{\zeta_j + 1} \tag{12}$$

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.

V. EXPERIMENTS AND RESULTS

A. Data Preparation

Existing standard IR test collections such as those used in TREC and 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.

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.

B. 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 [31] 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 [5] and [32]. 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 [1]. 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 [33]. Therefore, ranking from conceptually simple to advanced fits most of the users. As stated previously in [5] and [32], the authors also ranked documents from introductory to advanced when they tested their model on users possessing average level of knowledge about healthcare. In [34], 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 compo-

[1]http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stoplist/english.stop

[2]http://www.inex.otago.ac.nz/tracks/adhoc/gtd.asp

[3]http://www.seg.rmit.edu.au/zettair/index.html

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 II
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 12 SO THAT EQUAL WEIGHTS COME FROM BOTH COMPONENTS OF OUR MODEL.

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 III
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.

nents. The value of k in k-means was 150. We have set $f = 200$ (defined in Section III-B) because in general low number of factors are ideal for effective results [35]. We used SeDuMi with YALMIP [36] to conduct optimization in Equation 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 [31], 2) Dirichlet smoothed, query likelihood language model [37] (denoted as LM) with default parameter as in Zettair), and 3) Cosine similarity based retrieval [25]. In addition, we also compared with widely used unsupervised readability scores, namely, 1) ARI: Automated Readability Index, 2) Coleman-Liau (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 [10]. 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 CHM described in [7]. 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 [5] and [6] use an extra lexicon of technical terms.

C. 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 I. 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 and we use same formula as in [38]. 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 I. If NDCG is high, it means that the ranking function correlates better with the human judgments.

VI. RESULTS DISCUSSION

We present the main result in Tables II and III. Our model has significantly outperformed (using paired t-test $p < 0.05$) traditional ranking functions in Table II 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 [16] where the authors found out that stopwords have played an important role in their FAMCLASS classifier.

In Table III 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 IV, we present query-wise performance of our model compared with the comparative 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.

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 IV
QUERY-WISE PERFORMANCE OF OUR MODEL COMPARED WITH THE COMPARATIVE MODELS.

We experimented STTM by varying $0 \leq \beta \leq 1$ in Equation 12. We show results in Figure 2. 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 2 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.

VII. CONCLUSIONS AND FUTURE WORK

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. This finding is consistent with some prior works on document readability.

In future we will study the hyperlinked structure of the web and its role in determining conceptual difficulty of documents. The notion is that many simple web documents tend to link with other simpler documents and vice versa [39].

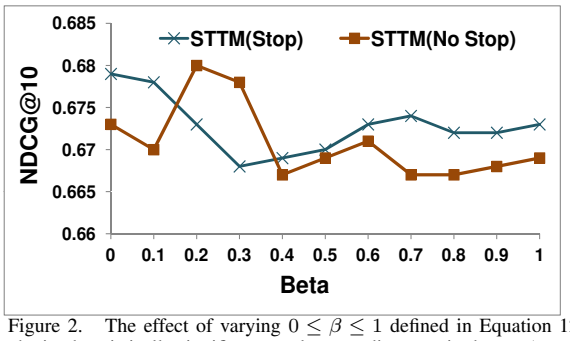

Figure 2. The effect of varying $0 \leq \beta \leq 1$ defined in Equation 12. We obtained statistically significant results according to paired t-test ($p < 0.01$) against all comparative methods.

REFERENCES

[1] S. K. Bhavnani, "Domain-specific search strategies for the effective retrieval of healthcare and shopping information," in *Human factors in Computing Systems*, 2002, pp. 610–611.

[2] A. Broder, "A taxonomy of web search," *SIGIR Forum*, vol. 36, no. 2, pp. 3–10, 2002.

[3] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by Latent Semantic Analysis," *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[4] C. Tan, E. Gabrilovich, and B. Pang, "To each his own: personalized content selection based on text comprehensibility," in *Proc. of WSDM*, 2012, pp. 233–242.

[5] X. Yan, D. Song, and X. Li, "Concept-based document readability in domain specific information retrieval," in *Proc. of CIKM*, 2006, pp. 540–549.

[6] J. Zhao and M.-Y. Kan, "Domain-specific iterative readability computation," in *Proc. of JCDL*, 2010, pp. 205–214.

[7] S. Jameel, W. Lam, C.-m. Au Yeung, and S. Chyan, "An unsupervised ranking method based on a technical difficulty terrain," in *Proc. of CIKM*, 2011, pp. 1989–1992.

[8] S. Jameel, W. Lam, X. Qian, and C.-m. Au Yeung, "An unsupervised technical difficulty ranking model based on conceptual terrain in the latent space," in *Proc. of JCDL*, 2012, pp. 351–352.

[9] M. W. Berry, S. T. Dumais, and G. W. O'Brien, "Using Linear Algebra for intelligent information retrieval," *SIAM Review (SIREV)*, vol. 37, no. 4, pp. 573–595, 1995.

[10] W. H. Dubay, "The principles of readability," *Costa Mesa, CA: Impact Information*, 2004.

[11] K. Collins-Thompson and J. Callan, "Predicting reading difficulty with statistical language models," *Journal of the American Society for Information Science and Technology*, vol. 56, no. 13, pp. 1448–1462, 2005.

[12] B. Bruce, A. Rubin, and K. S. Starr, "Why readability formulas fail," *IEEE Transactions on Professional Communication*, pp. 50–52, 1981.

[13] M. A. K. Halliday and R. Hasan, *Cohesion in English (English Language)*. Longman Pub Group, 1976.

[14] M. Bendersky, W. B. Croft, and Y. Diao, "Quality-biased ranking of web documents," in *Proc. of WSDM*, 2011, pp. 95–104.

[15] S. E. Schwarm and M. Ostendorf, "Reading level assessment using support vector machines and statistical language models," in *Proc. of ACL*, 2005, pp. 523–530.

[16] G. Kumaran, R. Jones, and O. Madani, "Biasing web search results for topic familiarity," in *Proc. of CIKM*, 2005, pp. 271–272.

[17] M. Heilman, K. Collins-Thompson, and M. Eskenazi, "An analysis of statistical models and features for reading difficulty prediction," in *Proc. of EANL*, 2008, pp. 71–79.

[18] E. Pitler and A. Nenkova, "Revisiting readability: a unified framework for predicting text quality," in *Proc. of EMNLP*, 2008, pp. 186–195.

[19] R. J. Kate, X. Luo, S. Patwardhan, M. Franz, R. Florian, R. J. Mooney, S. Roukos, and C. Welty, "Learning to predict readability using diverse linguistic features," in *Proc. of COLING*, 2010, pp. 546–554.

[20] X. Liu, W. B. Croft, P. Oh, and D. Hart, "Automatic recognition of reading levels from user queries," in *Proc. of SIGIR*, 2004, pp. 548–549.

[21] J. Y. Kim, K. Collins-Thompson, P. N. Bennett, and S. T. Dumais, "Characterizing web content, user interests, and search behavior by reading level and topic," in *Proc. of WSDM*, 2012, pp. 213–222.

[22] K. Collins-Thompson, P. N. Bennett, R. W. White, S. de la Chica, and D. Sontag, "Personalizing web search results by reading level," in *Proc. of CIKM*, 2011, pp. 403–412.

[23] R. Jones, R. Kumar, B. Pang, and A. Tomkins, ""I know what you did last summer": query logs and user privacy," in *Proc. of CIKM*, 2007, pp. 909–914.

[24] J. Morris and G. Hirst, "Lexical cohesion computed by thesaural relations as an indicator of the structure of text," *Computational Linguistics*, vol. 17, no. 1, pp. 21–48, 1991.

[25] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[26] J. Bellegarda, "Large vocabulary speech recognition with multispan statistical language models," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 76 –84, 2000.

[27] D. Beeferman, A. Berger, and J. Lafferty, "Statistical models for text segmentation," *Machine Learning*, vol. 34, pp. 177–210, 1999.

[28] W. Xu, X. Liu, and Y. Gong, "Document clustering based on Non-negative Matrix Factorization," in *Proc. of SIGIR*, 2003, pp. 267–273.

[29] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[30] W. Kintsch, "The role of knowledge in discourse comprehension: A construction-integration model," *Psychological Review*, vol. 95, pp. 163–182, 1988.

[31] S. Robertson, S. Walker, S. Jones, M. Hancock-Beaulieu, and M. Gatford, "Okapi at trec-3," 1996, pp. 109–126.

[32] M. Nakatani, A. Jatowt, and K. Tanaka, "Adaptive ranking of search results by considering user's comprehension," in *Proc. of ICUIMC*, 2010, pp. 27:1–27:10.

[33] R. W. White, S. T. Dumais, and J. Teevan, "Characterizing the influence of domain expertise on web search behavior," in *Proc. of WSDM*, 2009, pp. 132–141.

[34] X. Yan, R. Y. Lau, D. Song, X. Li, and J. Ma, "Toward a semantic granularity model for domain-specific information retrieval," *ACM Transactions on Information Systems*, vol. 29, no. 3, pp. 15:1–15:46, 2011.

[35] S. T. Dumais, "Latent semantic indexing (lsi): Trec-3 report," in *Overview of the Third Text REtrieval Conference*, 1995, pp. 219–230.

[36] J. Lofberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, 2004, pp. 284 –289.

[37] C. Zhai and J. Lafferty, "A study of smoothing methods for language models applied to information retrieval," *ACM Transactions on Information Systems*, vol. 22, no. 2, pp. 179–214, 2004.

[38] P. Cai, W. Gao, A. Zhou, and K.-F. Wong, "Relevant knowledge helps in choosing right teacher: active query selection for ranking adaptation," in *Proc. of SIGIR*, 2011, pp. 115–124.

[39] K. Akamatsu, N. Pattanasri, A. Jatowt, and K. Tanaka, "Measuring comprehensibility of web pages based on link analysis," in *Proc. of WI-IAT*, vol. 1, 2011, pp. 40–46.