Domain-Specific Iterative Readability Computation

Jin Zhao
Department of Computer Science
School of Computing
National University of Singapore
Singapore, 117590
zhaojin@comp.nus.edu.sg

Min-Yen Kan
Department of Computer Science
School of Computing
National University of Singapore
Singapore, 117590
kanmy@comp.nus.edu.sg

ABSTRACT

We present a new algorithm to measure domain-specific readability. It iteratively computes the readability of domain-specific resources based on the difficulty of domain-specific concepts and vice versa, in a style reminiscent of other bipartite graph algorithms such as Hyperlink-Induced Topic Search (HITS) and the Stochastic Approach for Link-Structure Analysis (SALSA). While simple, our algorithm outperforms standard heuristic measures and remains competitive among supervised-learning approaches. Moreover, it is less domain-dependent and portable across domains as it does not rely on an annotated corpus or expensive expert knowledge that supervised or domain-specific methods require.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; J.2 [**Computer Applications**]: Physical Sciences and Engineering

General Terms

Algorithm, Measurement

Keywords

Readability Measure, Iterative Computation, Domain-Specific Information Retrieval, Graph-based Algorithm

1. INTRODUCTION

The collections of domain-specific (*e.g.*, math and medical) resources available online have grown substantially over the years. Nowadays, there are not only large commercial web sites, such as paper databases and encyclopedias, but also millions of smaller web sites devoted to discuss domain-specific topics and/or their related resources [27]. As such, more people have incorporated internet search as part of their information seeking process for information and resources in a specific domain. For example, there are students looking for the definition of a math concept, professors searching for academic articles, patients trying to find health information and nurses seeking for evidence to support their practices. While these information needs can be satisfied by the resources, the process itself can be challenging. The main reason behind is that domain-specific resources target varying audiences, giving lexical evidence to distinguish different levels. For example, modular arithmetics can be explained in the context of ring theory or disguised as clock arithmetic, while the terminology used to describe symptoms of bird flu can be much more technical in a research article than a health information webpage meant to be accessible to laymen. However, most common search engines do not provide any indication of readability for the search results or allow for readability-based ranking. As a result, even though there are many relevant documents in the search results, users still spend a lot of time figuring out which ones are suitable to their level. This is especially true for the medical domain where the majority of health information are too difficult for the patients [9].

The key to the solution of this problem is an accurate readability measure for domain-specific resources. Although there are already quite a number of heuristic readability measures and supervised-learning approaches for readability measurement, they employ only generic text features, such as average length of words, part-of-speech and discourse relations. As a result, they are largely ignorant of the domain-specific elements (*i.e.*, concepts) present in domain resources and thus unable to measure their readability accurately. This is compounded by the fact that constructing the annotated corpus required for supervised approaches can be costly as well. In contrast, current domain-specific readability measures attempt to handle domain-specific concepts but largely with hand-built expert knowledge. For example, they use annotated familiarity scores to approximate difficulty or ontologies to obtain the genericness and associations of the domain-specific concepts. Despite the improvement on accuracy over the heuristic and supervised approaches, the major caveat is that such expert knowledge is still expensive and not easily available in most domains.

In this paper, we propose an algorithm for domain-specific readability computation which does not require an annotated corpus or expensive expert knowledge[1]. Our approach

JCDL'10, June 21–25, 2010, Gold Coast, Queensland, Australia.

[1]This work was partially supported by a National Research Foundation grant "Interactive Media Search" (grant R252-000-325-279).

is an iterative computation on a resource-concept graph based on the intuition that the readability of the domain-specific resources and the difficulty of domain-specific concepts provide accurate estimations of each other. Our evaluation has shown that this algorithm outperforms heuristic-based measures, remains competitive even among supervised-learning approaches, and is portable across domains.

The rest of the paper is organized as follows. We first review the relevant literature on readability research in Section 2. Then we describe our intuitions and the resulted iterative computation algorithm for domain-specific readability in Section 3. We evaluate our approach in the domain of math and medicine in Section 4 and discuss the directions for future research in Section 5. Lastly, we relate our algorithm to several graph-based iterative computation algorithms in Section 6 and make our conclusion in Section 7.

2. REVIEW OF READABILITY RESEARCH

Readability measures indicate how difficult it is to understand a piece of text. Therefore, they are commonly used by educators to select appropriate materials for the target audience. Although they have been applied in many different domains such as education, military and law, they are mostly generic, *i.e.*, without the flexibility to allow them to handle the special elements in any domain. Only recently have researchers started working on domain-specific readability measures. In the following sections, we first review two major classes of generic readability measures: the heuristic-based measures and the supervised-learning approaches, and then move on to the domain-specific ones.

2.1 Heuristic Readability Measures

According to a comprehensive review on classic readability studies [6], heuristic readability measures were first devised in the 1920s to facilitate the selection of textbooks. They are usually expressed as a weighted sum of the values of some features extracted from a piece of text. The features extracted are the ones that correlate well with readability while their weights are computed by linear regression.

Among all the text features, word features are considered as the strongest predictor. As early as 1923, Live and Pressey [18] already demonstrated that the median frequency of the words in a document correlated well with readability. Since then, word features have always been a staple in heuristic readability measures. For example, Vogel and Washburne [24] used the number of different words and the number of uncommon words, while Gray and Leary [10] employed the number of different unfamiliar words.

Other features have been considered as well: Vogel and Washburne [24] also examined five other classes of features, including sentence structure, part of speech, paragraph construction (*e.g.*, the number of sentences), general structure (*e.g.*, the number of lines in a book) and physical makeup (*e.g.*, weight and size of type). However, among these features, only the number of prepositions and number of simple sentences were found useful. Gary and Leary [10] further expanded the exploration of features by examining 64 countable variables in four categories: content, style, format and features of organization. They identified average sentence length, number of pronouns and number of prepositional phrases as useful in addition to word features.

In 1948, two most succinct yet reliable readability measures were devised: the Flesch-Kincaid Reading Ease (FKRE) formula [8] and the Dale-Chall readability formula [5]. Both consist of one sentence feature and one word feature. They share the average sentence length as the sentence feature but use the average number of syllables per word and the percentage of words out of a predefined list of 3,000 easy words respectively as the word feature.

Most of the readability formulas that come afterwards only simplify the computation process. For example, the Automated Readability Index (ARI) [14] and Coleman-Liau Index [3] count the characters in a word instead of the syllables, while Simple Measure of Gobbledygook (SMOG) [19] uses the number of polysyllable (*i.e.*, more than three syllables) words as the only feature. Therefore, up to today, the FKRE and Dale-Chall formulas still stand as the state-of-the-art heuristic readability measures.

Although heuristic readability measures provide a quick and indicative way to compute readability, they use only a small number of features to summarize the characteristics of a piece of text. This is often an oversimplification, as much information, such as the identity of the individual words and the knowledge encoded in the text, is lost in the process.

2.2 Supervised Learning Approaches

From the perspective of supervised learning, readability measurement can be viewed as a classification problem. For this method, one needs to first define a set of labels representing different levels of readability and use them to annotate a corpus of text documents as the training data. Once collected, features can be extracted from the training data to build a model that captures the relationship between the features and the labels. Then the resulted model can be used to predict the label of an unseen document based on its extracted features. The readability of this document is the readability level represented by the label.

Under this framework, many researchers have re-examined the utility of most text features. Starting from word features, Collins-Thompson and Callan [4] construct one unigram language model for each of the 12 American grade levels based on a corpus of webpages with grade-level annotations. These language models capture the probability of a word occurring in the document of a certain grade level. The readability of a new document is then predicted by finding the language model that most likely generates all the words in it. Their evaluation shows that this approach outperforms the traditional reading measure on webpages. Similarly, Leroy *et al.* [17] has adopted this approach in classifying the readability of health information into three levels (basic, intermediate and advanced), achieving a high accuracy of 98%. Further along this line, Schwarm and Ostendorf [22] explore the effect of using higher order n-gram models (up to trigram) on classification performance and show that it helps to minimize error rates.

Besides using higher order n-gram models, Schwarm and Ostendorf [22] also attempt to combine word features with other text features. They first compute the perplexity scores which indicate how well the language model of the document to be classified matches with the ones of the 12 grade levels. These perplexity scores are then used as the feature set of an Support Vector Machine (SVM) classifier together with other text features, such as FKRE score and out of vocabulary rate scores, as well as four parse features, such as average parse tree height and average number of noun phrases. Although the set of non-word features considered

is not large, this classifier is able to further minimize the error rates compared to the one based on trigrams.

An alternative approach to combine different types of features is to train one classifier for each type and then fuse their predictions. For example, Heilman *et al.* [12] extend the work of Collins-Thompson and Callan's [4] by introducing a k-Nearest Neighbour (kNN) classifier on grammatical features such as the sentence length and the patterns of the parse tree. The predictions from the kNN classifier are interpolated with the ones from the SVM classifier to produce a final prediction, which is better than using either one of the classifiers alone.

Most recently, Pitler and Nenkova [21] examine by far the largest set of textual features. Their feature set includes word (unigram language model), syntactic (identical to the parse features in Schwarm and Ostendorf's work [22]), lexical cohesion (*e.g.*, average cosine similarity between sentences), entity coherence (*e.g.*, the transition probability of an entity from being the subject in one sentence to the object in the next) and discourse relations (*i.e.*, language model over discourse relations instead of words). Their result shows that word features and average sentence length are strong predictors but the strongest ones are discourse features. Moreover, there is also a complex interplay between different types of features. While successful, their study is a proof-of-concept; they acknowledge that automatic extraction for such rich features does not yet exist.

Despite the fact that supervised learning approaches offer better accuracy compared to heuristic measures, there are still two main issues that limit their utility in domain-specific readability measurement: First, all previous work require an annotated corpus as the training data. This is costly to construct for domain-specific resources, whose annotation can only be done by experts. Second, although language modeling helps to generate useful word features, it is largely ignorant of the domain-specific concepts. In other words, it treats domain-specific concepts as a sequence of tokens without considering their semantics or the relationships among them. Therefore, it would not be as effective for domain-specific readability measurement.

2.3 Domain-Specific Readability Measures

To reduce the need for a corpus and better handle domain-specific concepts, domain-specific readability measures have focused on identifying the difficulty of such concepts with expert knowledge. Depending on the type of expert knowledge utilized, these measures can be classified into two categories: wordlist-based and ontology-based approaches.

2.3.1 Wordlist-based Approaches

The wordlist-based approaches derive the conceptual difficulty of domain-specific concepts from domain wordlists. For example, in the domain of consumer healthcare, Kim *et al.* [13] use the average term and concept familiarity scores from the Open Access and Collaborative Consumer Health Vocabulary (OAC CHV) as the the difficulty of terms and concepts. A distance score is computed based on how far an unseen document differs from known document samples. This score is combined with two other distance scores based on text length and syntactic features to become the final readability measure. This approach is able to correlate well with the heuristic-based measures on most documents, while correctly identifying the difficult documents which heuristic-based measures miss. However, whether the familiarity features work well compared to the other features is left unexamined in this study.

Borst *et al.* [1] associate the conceptual difficulty with rarity. This is in turn estimated by the size of generic English wordlists (12,000 to 264,000) in which a medical term appears. Their hypothesis is that the smaller the wordlist a word appears in, the more common (and thus less difficult) it is. The complexity of the words in a document is summarized by their average complexity and combined with the average sentence length to produce a final score. An accuracy of 92% is achieved when applied to the two case problem of distinguishing documents targeted at non-experts from the ones targeted at medical professionals.

2.3.2 Ontology-based Approaches

Different from the wordlist-based approaches, the ontology-based approaches utilize an existing ontology of domain-specific concepts to derive possible indicators for readability. Yan *et al.* [26] introduce two additional components into the Dale-Chall Readability formula for medical documents: document scope and document cohesion. The document scope is based on the scope of the medical terms in the document, which is in turn defined as their depth in the Medical Subject Heading[2] (MeSH) hierarchy. On the other hand, the document cohesion measures the relatedness of the medical terms in a document. The more associations the terms have in the ontology, the more cohesive a document is. The combined formula is reported to be significantly better correlated with the readability of the medical documents, when compared to heuristic readability measures.

In short, these measures address two issues of supervised learning approaches: the need for a corpus and ignorance of domain-specific concepts. However, they still require expert knowledge and incur substantial labor cost in constructing their annotated wordlist or ontology. These resources may not be available for other domains. As a result, the applicability of such methods remains limited.

2.4 Summary

The study in heuristic readability measures has identified word difficulty and average sentence length as the two important readability indicators among all the text features. This is enhanced by the supervised learning approaches which have enabled deeper text features to be extracted automatically and combined with more sophisticated statistical models. In spite of the better accuracy achieved, they require annotated domain-specific corpora and are largely ignorant of the domain-specific concepts. Domain-specific readability measures address the two issues by deriving information from domain-specific concepts using expert knowledge; however, the cost and availability of expert knowledge still limit the applicability of such approaches.

All previous works have refined generic readability measures to be sensitive to nuances within a domain by using annotated resources. Is there a way to introduce domain-specific readability without the use of expensive supervision?

Our method addresses this need. Similar to other domain-specific measures, it derives further information (*i.e.*, difficulty) from a list of domain-specific concepts and it as a indicator for readability. However, this is done without any annotated corpus or expensive knowledge source. Therefore,

[2]http://www.nlm.nih.gov/mesh/

our approach is able to provide better readability estimation for domain-specific resources compared to generic readability measures, and, more importantly, can be ported across a wide variety of domains.

3. METHODOLOGY

In a nutshell, our method first constructs a bipartite graph with two sets of nodes representing domain-specific resources and domain-specific concepts, respectively, and edges representing the occurrence of the latter in the former. Then we iteratively compute 1) the readability score for a resource node based on the difficulty scores of the adjacent concept nodes, and 2) the difficulty score for a concept node based on the readability scores of the adjacent resource nodes.

The required inputs for our algorithm are a list of domain-specific concepts and a corpus of domain-specific resources. A key distinction of our work from previous works is that both do not need to be annotated – a flat list of concepts and a corpus of resources is all that is required. Note that "resources" here connote any textual resource (*e.g.*, an scholarly article, webpage, formalized educational lesson module, or a newspaper clipping), but in the context of this paper, we occasionally use "webpages" or "documents" to stand in for the more general notion of "resources".

These are easy requirements to satisfy for most domains: A list of domain-specific concepts is usually available in the form of a domain-specific dictionary, encyclopedia, or the index at the back of a textbook. Given such a list, a domain-specific corpus can be constructed by downloading the top N (*e.g.*, 100) results of each of the listed concepts from a search engine. Conversely, if a list of domain-specific concepts cannot be found but there are existing collections of domain-specific resources, such collections can be taken directly as the corpus while the list can be constructed by extracting key phrases [25] or by simply listing all the noun phrases from it. Lastly, if neither of them exists, one can manually select a small number of domain-specific concepts as a seed list, and then collect a corpus of domain-specific webpages with the help a search engine. One can then iteratively expand them by extracting phrases from the corpus to expand the list and then using the expanded list to collect more webpages for the corpus.

In any case, the amount of expert knowledge needed (*i.e.*, knowing whether a concept belongs to a specific domain) is significantly less than the amount needed by other domain-specific readability measures (*i.e.*, understanding the concepts sufficiently well to assign a score or construct an ontology out of them). Therefore, we consider our approach to be less dependent on domain-specific knowledge sources and more portable across domains.

We will first explain the intuition behind our method and then describe the algorithm for the computation in detail.

3.1 The Intuition

Our method is based on a simple mutually recursive observation between domain-specific resources and concepts:

- A domain-specific resource A is less readable than another domain-specific resource B if A contains more difficult domain-specific concepts than B.
- A domain-specific concept A is more difficult than another domain-specific concept B if A is mentioned in less readable domain-specific resources than B.

This intuition helps us solve cases where the generic readability measures lead to incorrect conclusions for the difficulty of domain-specific concepts in isolation. For example, let us say we need to determine whether a resource containing the concept 'ring theory' is less readable than another one containing the concept 'Pythagorean theorem'. If we extract normal text features such as the average number of syllabus or the percentage of familiar words, 'Pythagorean theorem' would be incorrectly calculated as more difficult than 'ring theory'. However, if we examine a corpus of resources containing these two words, we may discover that 'ring theory' also appears on less readable pages about advanced math concepts, such as 'isomorphism theorem' and 'Abelian group', whereas 'Pythagorean theorem' appears on more readable pages about basic math concepts, such as 'triangle' and 'sine'. With this information, we can decide that 'ring theory' is more difficult than 'Pythagorean theorem' and infer that the resource containing 'ring theory' are less readable than one containing 'Pythagorean theorem'.

In this way, we can determine the relative readability of domain-specific resources by the relative difficulty of the domain-specific concepts they contain and vice versa. In the web context, we use concepts as the context for assessing resource readability, and webpages as the context for assessing concept difficulty.

3.2 The Algorithm

We first construct a resource-concept graph. This graph is bipartite, containing two types of nodes, one representing concepts, the other representing resources. Edges are added between a concept node and a resource node to represent the occurrence of former on the latter. After constructing this graph, we start the score computation by first assigning an initial difficulty score to each concept node and a readability score to each resource node. We can then iteratively update the readability scores for the resources based on the difficulty scores of the associated concepts (and vice versa) until the termination condition is met. The final scores at the nodes can be taken as the readability for resources and difficulty for concepts. We describe the details of graph construction and score computation in the following sections.

3.2.1 Graph Construction

Given a list of concepts and a collection of resources, the first and most important step in constructing the page-concept graph is to count the occurrences of the concepts in the resources. To do so, we first index all the resources using the open source text search engine Lucene[3]. We then use each of the concepts as a query to retrieve a set of matching resources. Lastly, we derive the number of occurrences from the term frequency vectors of the retrieved resources.

With the occurrence statistics collected, the construction of the graph is straightforward (Algorithm 1): we create a representing concept node for each concept in the list (Lines 2-4) and a representing resource node for each resource in the collection (Lines 5-7). We then add an edge between an concept node and a resource node if the concept represented by the former occurs on the resource represented by the latter (Lines 8-11). This completes the construction of graph and Fig.1 gives an example of a graph constructed based on two resources and a list of concepts.

[3]http://lucene.apache.org/

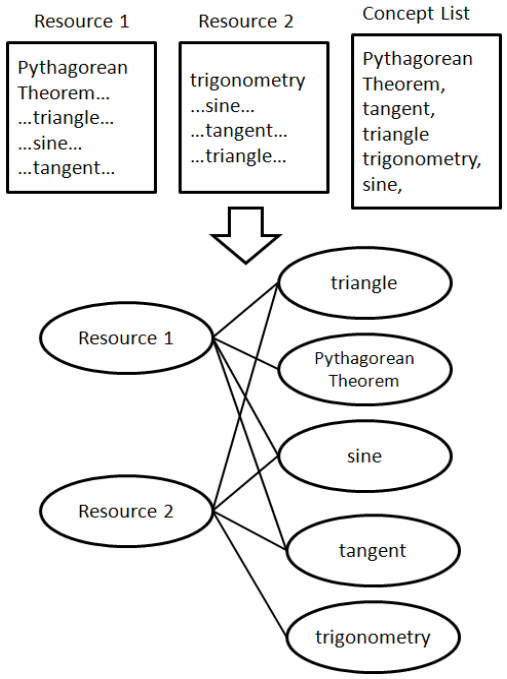

Figure 1: Example of Graph Construction

Algorithm 1 constructGraph($concept-list$, $corpus$)

```
1: initialize empty graph G
2: for all concept c in concept − list do
3:    create a new concept node cNode representing c
4:    add cNode to G
5: for all resource r in corpus do
6:    create resource node rNode representing r
7:    add rNode to G
8: for all resource node rNode in G do
9:    for all concept node cNode in G do
10:      if the concept c represented by cNode appears in
         the resource r represented by rNode then
11:         add edge (rNode, cNode) to G
12: return G
```

3.2.2 Score Computation

The score computation starts with an initialization step (Algorithm 2). In this step, we assign an initial score to each resource node (Lines 1-2), representing its readability, and each concept node (Lines 3-10), representing its difficulty:

To initialize the score for a resource node, we choose to use the FKRE formula as it is one of the classic, widely-used heuristic readability formula as described in Section 2.1.

$$score(rNode) = 206.835 - 1.015 * avgSL(r) - 84.6 * avgWL(r),$$

where $avgSL(r)$ and $avgWL(r)$ stand for the average sentence length in words and the average word length in syllables of the resource r respectively [8].

For a concept node, we initialize its score as the average readability of all the resources containing the concept.

$$score(cNode) = \frac{\sum_{rNode \in Adj(cNode)} score(rNode)}{|Adj(cNode)|},$$

Algorithm 2 initialize(G)

```
1: for all resource node rNode in G do
2:    score(rNode) ← readability(r)
3: for all concept node cNode in G do
4:    score(cNode) ← 0
5:    counter ← 0
6:    for all resource node rNode in G do
7:       if edge (rNode, cNode) exists in G then
8:          score(cNode) ← score(cNode) + score(rNode)
9:          counter ← counter + 1
10:   score(cNode) ← score(cNode)/counter
```

Algorithm 3 iterate(G)

```
1: for all node n in G do
2:    newScore(n) ← 0
3:    counter ← 0
4:    for all node aNode in G do
5:       if edge (n, aNode) exists in G then
6:          newScore(n) ← newScore(n) + score(aNode)
7:          counter ← counter + 1
8:    newScore(n) ← newScore(n)/counter + score(n)
```

where $Adj(cNode)$ stands for the collection of nodes adjacent to $cNode$.

We then proceed to the iterative computation step in which the new score of each node is as the average of the scores of the neighboring nodes plus its current score (Algorithm 3):

$$newScore(n) = \frac{\sum_{aNode \in Adj(n)} score(aNode)}{|Adj(n)|} + score(n),$$

where $Adj(n)$ stands for the collection of nodes adjacent to the node n.

After each iteration, we check whether the termination condition is met (Algorithm 4). This is done by computing the change in the ranks of the resource nodes based on their scores to see if it stabilizes. We take the square root of the residual sum of squares (RSS) divided by the number of nodes as a measure for the change. More specifically, the change is computed using the following formula:

$$change = \sqrt{\frac{(\sum_{rNode \in G} (newRank(rNode) - rank(rNode))^2)}{totalNumberOfRNodes}}.$$

If the change in the ranks stabilizes (*i.e.*, is smaller than the threshold), the scores of the nodes will be updated a final time as the new scores and the computation terminates; otherwise, the update is followed by more iterations until the termination condition is finally met. Upon termination, the scores of the concept nodes and the resource nodes are, respectively, the computed difficulty and readability scores. Note that the values of the scores themselves are not of import, but rather that the relative order between individual concept or resource.

In the case where new resources and concepts are added after the iterative computation is completed for the existing resource collection and concept list, we can update the graph structure accordingly, initial the scores of the newly added nodes as the average score of their adjacent nodes, and then carry out further iterative computations on the

Algorithm 4 terminate(G)

$change \leftarrow 0$
$counter \leftarrow 0$
$RSS \leftarrow 0$
for all resource node $rNode$ in G **do**
  $newRank(rNode) \leftarrow convert(newScore(rNode))$
  $rank(rNode) \leftarrow convert(score(rNode))$
  $RSS \leftarrow RSS + (newRank(rNode) - rank(rNode))^2$
  $counter \leftarrow counter + 1$
$change \leftarrow (RSS/counter)^{1/2}$
return $(change < THRESHOLD)$

updated graph until the termination condition is (again) met. Alternatively, we may rerun the algorithm on the enlarged resource collection and concept list. This should provide more accurate estimation especially when the number of newly added resources and concepts is substantial.

There are already a number of well-established algorithms in the web search domain for computing quality scores for webpages such as PageRank, HITS, and SALSA. However, as far as we know, our work is the first to apply this methodology for domain-specific readability measurement. We will relate our approach to the the existing graph-based iterative computation algorithms in Section 6.

4. EVALUATION

We have two specific goals we wish to achieve in evaluating our algorithm. Our primary goal is to demonstrate the efficacy of our approach and our secondary goal is to demonstrate our technique's domain independence.

To accomplish these two goals, we performed three sets of experiments in two different domains. As our funded project work centers on the domain of mathematics, we carried out a set of experiments to measure the performance of our approach with a collection of math resources and concepts and compared it to four different baselines. Second, since a truly domain independent method should rely on as little domain-specific resources and concepts as possible, we have also investigated into how many math resources and concepts our method needs to achieve good performance. Last, we evaluated the performance of our approach on medical documents to show its portability across domains. We discuss these evaluations in turn.

4.1 Experiment in the Math Domain

While our technique is minimally supervised, to properly assess the results, we need to first compile a set of materials that have gold-standard readability annotations. To ensure fairness, we sought additional annotators for our main mathematics corpus. The resulting construction, annotation and validation of the ground truth took 3 man-months. We feel that this was a significant investment of resources and would be a data bottleneck for other comparative work. As such, to encourage comparative work, we have made the resulted corpus and judgments available for download[4].

We follow our own earlier recommendations in corpus collection. We use a corpus of mathematically related webpages extended from our earlier work [27] for the evaluation of our algorithm. In total, we have chosen 27 common math concepts from MathWorld encyclopedia, covering different aspects of math, such as areas (*e.g.*, "geometry" and "number theory"), operations (*e.g.*, "Fourier transform"), theorems (*e.g.*, "Pythagorean theorem") and objects (*e.g.*, "complex number"). We chose them specifically to reflect the diversity of concepts in math and ensure the webpages collected cover a wide spectrum of readability.

[4]http://wing.comp.nus.edu.sg/downloads#mwc

Table 1: Readability Scale for Webpages

Value	Corresponding Education Background
1	Primary
2	Lower Secondary
3	Higher Secondary
4	Junior College (Basic)
5	Junior College (Advanced)
6	University (Basic)
7	University (Advanced)

For each chosen math concept, we performed a Google web search and incorporated the first 100 results into our corpus. To obtain the ground truth readability judgments for evaluation, we asked 30 undergraduate students to annotate the readability level for 120 randomly chosen, manually segmented, math relevant webpages from our corpus. Other dimensions of the webpages were also annotated, but the discussion of these dimensions are out of the scope of this study, and are not mentioned further. The details of the readability levels used can be found in Table 1.

Subjects were first shown an annotation guide explaining how to use our web-based annotation system and what the readability levels are. After reading the guide, the subjects annotated webpages by reading math webpages and selecting an appropriate readability level as shown in Fig. 2. Each subject was asked to annotate 20 webpages in 45 minutes and were given a token amount as appreciation for their efforts. On average, each webpage was annotated by 5 to 8 subjects. We take the average annotated values to establish the ground truth of readability.

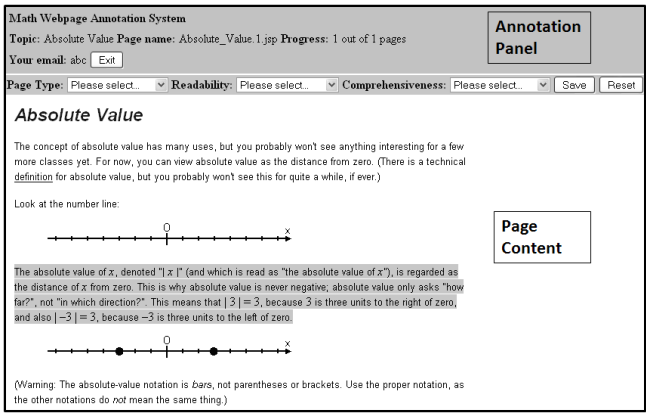

Figure 2: Webpage Annotation Interface. Subjects select a readability value for the webpage from the drop-down menu at the annotation panel.

Before the experiment, we also needed to determine whether manual readability annotation is indeed a feasible and reproducible task. To do this, we assessed inter-annotator reliability by first computing the pairwise inter-judge agreement using Cohen's Kappa coefficient [2]. Cohen's Kappa mea-

sures the agreement between two annotators, accounting for chance agreement. Its values range from 1.0 (complete correlation/agreement) to -1.0 (complete disagreement/negative correlation). A zero value indicates no correlation. The average pairwise inter-judge agreement was .72, indicating substantial agreement. We also applied Fleiss' Kappa [7], a multi-rater agreement measure, to calculate the agreement among all the subjects. The result was similar (.73).

The measured agreement was substantial but not strong (not above .8). We manually examined the annotations to discover which labels were being confused. We observed that although the subjects are able to determine what is readable and what not, the exact value annotated may still differ slightly between subjects. This is shown by the fact that 67% of the disagreed readability annotations had a standard deviation of less than 0.5. To eliminate these small perturbations, we applied Spearman's rho [23], which converts the values to rank order. The measured correlation is .93 (again, read on a -1.0 to +1.0 scale). This indicates a strong correlation for rank order and confirms our hypothesis that the general order of readability can be reliably distinguished.

As readability measures were devised to facilitate material selection, it is more important to be able to determine the relative importance between pairs of documents rather than assigning exact labels. Thus, we evaluate our approach by the pairwise judgments accuracy. For each pair of webpages in the collection, we examine their readability scores from the subjects and those from our system. A pairwise judgement is said to be correct if both scores agree on whether one is more (or less) readable than the other. We ignored pairs of annotated readability values whose difference are smaller than a threshold (0.5) – we considered such pairs indistinguishable even by our subjects – and hence not suitable to be included into evaluation. In total, there were 5,165 qualified pairwise judgements for the annotated webpages.

4.1.1 General Evaluation

We ran our system with all the webpages in our corpus and a list of math concepts compiled from MathWorld Encyclopedia. We present the pairwise judgment accuracy (as well as the Spearman's rho) of our system (denoted as IC) and the four baselines in Table 2. The best performance of our system after resource and concept selection (denoted as ICS, to be introduced later in Section 4.1.2) is also shown.

The four baselines include one standard heuristic measure (FKRE score) and three supervised learning approaches: Naïve-Bayes (NB) classifier[5], SVM classifier[6] and Maximum Entropy (Maxent) classifier[7]. The three classifiers are trained on the annotated webpages and use only binary features indicating whether a particular math concept appears on the webpage. We intentionally limited these baseline classifiers to use the same inputs as our IC method, as we are only interested in how well they could make use of the concepts to perform readability measurement. We also tried adding discretized versions of average word length, average sentence length and the FKRE score into the baselines' feature sets, but this did not manage to improve their performance. For all the supervised learning approaches, 5-fold cross validation was performed to avoid overfitting.

[5]http://www.cs.waikato.ac.nz/ml/weka/
[6]`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`
[7]http://maxent.sourceforge.net/

Table 2: Evaluation Results on Math Webpages

	Pairwise	Spearman
FKRE	.72	.48
NB	.72	.52
SVM	.80	.70
Maxent	.82	.67
IC	**.85**	**.72**
ICS	**.87**	**.77**

As can be seen from the table, FKRE showed a modest amount of correlation (.72 on pairwise judgment accuracy and .48 on Spearman's rho). This is similar to the results achieved by the NB classifier. In contrast, the two other baselines, SVM and Maxent, performed significantly better, scoring .80 and .82 on pairwise judgment accuracy and .70 and .67 on Spearman's rho respectively. However, our approach still outperformed all the baselines with .85 and .72 on the two metrics.

We believe these results strongly validate our method and address our first experimental goal. In order to verify the second goal of domain-independence, we have also run another set of experiments in the math domain to study how our method's performance varies when only a subset of the webpages and concepts is given.

4.1.2 Evaluation with Selection Strategies

In this set of experiments, we use only a subset of math webpages and concepts selected by four different selection strategies: 1) selecting N webpages at random, 2) selecting the top N webpages with the highest quality, as indicated by their ranks in the search results from which they were collected, 3) selecting N concepts at random, and 4) selecting the top N most important concepts as indicated by a concept-based version of TF.IDF. The N mentioned in the selection strategies is set to five different levels: 20%, 40%, 60%, 80% and 100%. The resulting performance of these selection strategies are shown in Fig. 3-4.

Two points are noteworthy from the results: First, as more webpages are selected, the performance of our system improves. However, the initial performance and the rate of improvement differ for the two page selection strategy. If we choose the webpages at random, the initial performance (N=20%) is much lower than the one achieved by choosing the webpages by quality. In contrast, when webpages are chosen by quality, the initial performance is only marginally worse than the one achieved using all the webpages. This shows that our method can work with a very small set of webpages as long as they are of high quality.

Second, similar to the results on page selection strategies, if concepts are chosen at random, increasing the number of concepts helps to improve the performance. However, if the concepts are chosen by importance, using only the top 60% of the concepts in fact further boosts the performance of our system. This indicates that the concepts with low TF.IDF do not contribute positively to the performance and should be removed from the graph using this selection strategy. Therefore, we have also incorporated the concept selection by TF.IDF into our system and denoted this improved version as ICS. The resulting performance is further improved to .87 and .77 on the two metrics as shown at the last row of Table.2.

In summary, our approach is able to work with a small set of domain-specific resources and concepts to achieve good performance with simple, automatic selection strategies. Therefore, it is highly portable to any domains even for the ones in which it is difficult to collect a sizeable collection of domain-specific resources or a list of domain-specific concepts.

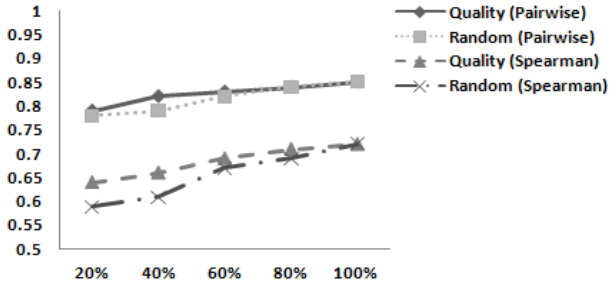

Figure 3: Effects of Page Selection Strategy

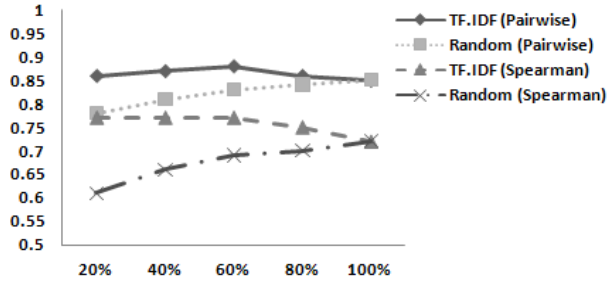

Figure 4: Effects of Concept Selection Strategy

4.1.3 *Error Analysis*

While we are satisfied with our approach, a detailed error analysis revealed two potential areas for improvement:

First, as we do not pre-process the webpages to identify the main content of the webpages, math concepts that are presented as auxiliary information, such as navigational links and advertisements, have added substantial noises to the graph construction process. For example, the math concept "number theory" happens to appear at the navigational panel from MathWorld. Consequently, all the 39 MathWorld pages in our corpus, which make up about 10 percent of the pages containing the math concept, are included into the difficulty computation for this concept and have adversely affected the accuracy of our approach. Moreover, there are also many webpages in our corpus whose main contents are quite readable, but whose "related concepts" section contains a large number of difficult math concepts (*e.g.*, there is an encyclopedic page on polynomial which lists more than 60 difficult concepts). In such cases, the computed readability is artificially inflated. We believe that further pre-processing to exclude certain sections of the webpages would significantly reduce the number of errors.

Second, in the current formulation, all concepts on a page are considered to be equal regardless of whether they are only briefly mentioned or explained in detail. Ideally, if a concept is only briefly mentioned, it should not be considered as very important for the page as well as in the readability computation. If the relative importance of concepts on a page can be determined, we can use a weighted average to suppress the unimportant ones and obtain a more accurate estimation of readability. We believe natural language analysis of the webpage would be needed to compute the relative importance and determine the weights automatically.

Table 3: Evaluation Results on Medical Webpages

	Pairwise	Spearman
Heuristic	.63	.28
NB	.73	.53
SVM	**.82**	**.70**
Maxent	.76	.60
IC	.72	.49
ICS	.75	.54

4.2 Experiment in the Medical Domain

Our experiment in the medical domain also followed the same general methodology. We first selected 27 medical concepts of varying difficulty levels from MeSH, covering different aspects of medicine, such as diseases (*e.g.*, cough), injuries (*e.g.*, bruise), substances (*e.g.*, vitamin), symptoms (*e.g.*, snoring), therapies (*e.g.*, blood transfusion) and procedures (*e.g.*, bronchoscopy). For each of these concepts, we then downloaded the top 100 search results and consolidated the webpages for our medical corpus. Due to budgetary limitations, this corpus was only manually annotated by the first author. Readability values were annotated for a subset of the corpus (946 pages) using the same labels.

4.2.1 *General Evaluation*

We ran our system with all the medical webpages and a list of medical concepts compiled from MeSH. The results are listed in Table 3. In this experiment, there are 320,976 pairwise judgments.

The performance of our approach for the medical domain is modest in comparison to the math domain. On one hand, our system still significantly outperformed the heuristic measures: pairwise judgement accuracy improves from .62 to .72 (.75 after concept selection, threshold = 40%) while Spearman's rho increases from .28 to .49 (.54 after concept selection) ($p < 0.001$ for all cases). On the other hand, when compared to the supervised classifiers, our approach achieved similar results as the the NB classifier and Maxent classifier but did not manage to outperform the SVM classifier. However, considering the fact that our approach did not have access to the large amount ($\sim$1000) of readability annotations as the supervised classifiers did, we consider our approach as performing reasonably well and believe that this test does demonstrate its portability.

4.2.2 *Error Analysis*

As for the potential sources of errors, we observe that the medical webpages contain more noise than the mathematical ones. This is because health information webpages are often commercial in nature and contain many advertisements which overwhelm the main content. In addition, they also tend to include more related medical concepts in navigation bars. For example, there is a webpage about snoring which lists more than 100 medical concepts at its navigation bar while its main content only contains less than 20. We believe this higher degree of noise is one of the factors that compromise the performance of our system. Nevertheless,

this should be readily solvable if we apply pre-processing to exclude certain sections beforehand.

Another factor that compromises our system is the more limited spectrum of readability levels in the medical corpus, in comparison to mathematics. Although we have intentionally chosen concepts of different difficulty levels and from different areas, medical concepts are inherently difficult. None of the webpages are targeted to primary school students. This is rather different from the math scenario, where we can easily find highly readable webpages full of games and animations that explain easy math concepts to younger audiences. Without such webpages, our algorithm is limited in its ability to discern and boost basic readability scores. This suggests that one can estimate the effectiveness of our algorithm in a particular domain by measuring the width of the readability spectrum. We hypothesize that the wider the spectrum, the more effective our algorithm will be. We plan to validate this hypothesis in future work.

5. DIRECTIONS FOR FUTURE RESEARCH

In addition to the ways to improve our algorithm as mentioned in Section 4.1.3 and 4.2.2, we also notice that there are cases where it is insufficient to use a single value for readability or difficulty for a concept. We believe this can be solved by modeling difficulty and readability using probability distributions. For example, we may be able to derive from the corpus that 70% of the pages related to the concept "geometry" are highly readable (*e.g.*, pages targeted at primary school students about simple geometric shapes), while the other 30% are much less readable (*e.g.*, pages targeted at university students on differential geometry). Then we can more accurately model the difficulty of "geometry" as 70% easy and 30% difficult instead of using the average value.

Moreover, our current approach models only the relation between one pair of attributes – difficulty and readability – between domain-specific resources and domain-specific concepts. However, there are still much more other correlations between other pairs of attributes as we have observed in our corpus. For example, books are commonly written for generic topics (*e.g.*, linear algebra). This is one of our observed cases where the "type" of the resources can be influenced by the "genericity" of concepts.

In order to properly model these considerations, we are developing a probabilistic framework with Bayesian Networks [11]. A Bayesian Network (BN) is a generic probabilistic framework that has been widely used for inference in many different domains. It is a directed acyclic graph with nodes representing variables and edges encoding conditional dependence. The algorithm discussed here is easily incorporated into this framework when we consider the constructed graph as a two-layer (concept and resource layers) BN and use parameter learning as the iterative computation step. In addition, the BN's probabilistic nature and genericity would allow us to use probability distribution to model any attributes and study their correlation easily.

6. RELATED GRAPH-BASED ITERATIVE COMPUTATION ALGORITHMS

Our work is inspired by other successful iterative graph algorithms which have made their impact in digital libraries. We relate and contrast our approach to three of them: PageRank, HITS and SALSA.

PageRank [20] is a link analysis algorithm based on the intuition that the number of backlinks of a webpage is a good indication of its popularity or importance. It works on a graph which contains nodes representing webpages (or publications or authors in a digital library) and the directed edges representing the link from the source to the target node (*e.g.*, a hyperlink or citation). The score of a node is computed as the probability of a random surfer visiting the corresponding node by following the hyperlinks. This algorithm has been very successful and widely used in domains such as web searches and citation analysis. However, in our problem, we need to model two types of objects, resources and concepts, in the graph. Correspondingly, the edges in our graph represent occurrences. Under our construction, having more links means there is a resource that has a higher number of different concepts or a concept that has a higher domain frequency. Due to the fact that the readability of a resource depends on the number of "difficult" concepts instead of the number of different concepts, while the difficulty of a concept tends to be inversely correlated with its domain frequency, we believe that a direct application of PageRank would not work for our problem.

HITS [15] is more similar to our algorithm compared to PankRank in the sense that it also keeps track of two separate *hub* and *authority* scores, using them to compute each other iteratively. The main difference between HITS and our approach is that we consider two types of objects and attach the two *difficulty* and *readability* scores separately. In addition, HITS constructs the graph online using a subset of documents from the corpus retrieved by a query, whereas our algorithm constructs the graph offline with all the documents in the collection.

SALSA [16] combines the strength of PageRank and HITS by incorporating the backlink information into the hubs and authority computation. However, the idea of using backlinks as an indication of readability or difficulty does not make much sense in our application.

7. CONCLUSION

We propose an iterative computation algorithm for domain-specific readability measurement based on the intuition that the readability of domain-specific resources and the difficulty of domain-specific concepts can be recursively estimated from each other.

As such, in our algorithm, we first construct a graph representing the resources and concepts, and then iteratively update 1) the readability score for a resource based on the difficulty scores of the domain-specific concepts it contains and 2) the difficulty score for a concept based on the readability scores of the resources which it appears in.

Our approach improves the accuracy of readability measurement over the standard heuristic measures and remains competitive among supervised learning approaches in both math and medical domain. Moreover, our approach only requires a list of domain-specific concepts and a corpus of domain-specific resources. These requirements are less strict and less domain-dependent compared to both supervised and domain-specific approaches which require an annotated corpus or expensive expert knowledge. Therefore, we believe that our approach is a simple yet effective and portable solution to measure the readability for domain-specific resources.

In the future, we plan to further improve our approach by constructing a probabilistic framework to generically model

the attributes of the domain-specific resources and concepts and the possible correlations between them.

Our research on readability is part of a long-term project towards developing domain-specific search engines, specifically for the math and medical communities. We plan to utilize such readability estimation techniques into appropriate browse/search interfaces, such that users will be able to locate domain-specific resources suitable to their level of expertise quickly. This would help to validate the utility of readability in domain-specific information retrieval and bring real benefits to end users.

8. REFERENCES

[1] A. Borst, A. Gaudinat, C. Boyer, and N. Grabar. Lexically based distinction of readability levels of health documents. In *MIE*, 2008.

[2] J. Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, April 1960.

[3] M. Coleman and T. Liau. A computer readability formula designed for machine scoring. *Journal of Applied Psychology*, 60(2):283–284, 1975.

[4] K. Collins-Thompson and J. P. Callan. A language modeling approach to predicting reading difficulty. In *HLT-NAACL*, pages 193–200, 2004.

[5] E. Dale and J. S.Chall. A formula for predicting readability: Instructions. *Educational Research Bulletin*, 27:37–54, 1948.

[6] W. DuBay. *Unlocking Language: The Classic Readability Studies*. BookSurge Publishing, 1990.

[7] J. L. Fleiss. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382, 1971.

[8] R. Flesch. A new readability yardstick. *Journal of Applied Psychology*, 32:221–233, 1948.

[9] R. C. Graber MA and K. B. Readability levels of patient education material on the world wide web. *Journal of Family Practice*, 48(1):58–61, 1999.

[10] W. S. Gray and B. Leary. *What makes a book readable*. Chicago Press, 1935.

[11] D. Heckerman. A tutorial on learning with bayesian networks. In *Learning in graphical models*, pages 301–354, Cambridge, MA, USA, 1999. MIT Press.

[12] M. Heilman, K. Collins-Thompson, J. Callan, and M. Eskenazi. Combining lexical and grammatical features to improve readability measures for first and second language texts. In *HLT-NAACL*, pages 460–467, 2007.

[13] H. Kim, S. Goryachev, G. Rosemblat, A. Browne, A. Keselman, and Q. Zeng-Treitler. Beyond surface characteristics: A new health text-specific readability measurement. In *AMIA*, 2007.

[14] J. P. Kinkaid. Use of the automated readability index for evaluating peer-prepared material for use in adult reading education. Technical report, Georgia Southern Coll., Statesboro, 1972.

[15] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[16] R. Lempel and S. Moran. The stochastic approach for link-structure analysis (salsa) and the tkc effect. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 33(1-6):387–401, 2000.

[17] G. Leroy, T. Miller, G. Rosemblat, and A. Browne. A balanced approach to health information evaluation: A vocabulary-based naïve-bayes classifier and readability formulas. *Journal of the American Society for Information Science and Technology*, 2008.

[18] B. A. Lively and S. L. Pressey. A method for measuring the 'vocabulary burden' of textbooks. *Educational Administration and Supervision*, 9:389–398, 1923.

[19] H. G. McLaughlin. SMOG grading - a new readability formula. *Journal of Reading*, 12(8):639–646, May 1969.

[20] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library Technologies Project, 1998.

[21] E. Pitler and A. Nenkova. Revisiting readability: A unified framework for predicting text quality. In *EMNLP*, pages 186–195, 2008.

[22] S. E. Schwarm and M. Ostendorf. Reading level assessment using support vector machines and statistical language models. In *ACL '05*, pages 523–530, 2005.

[23] C. Spearman. The proof and measurement of association between two things. *The American Journal of Psychology*, 100(3-4):441–471, 1987.

[24] M. Vogel and C. Washburne. An objective method of determining grade placement of children's reading material. *The Elementary School Journal*, 28:373–381, 1928.

[25] I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and Craig. Kea: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255, 1999.

[26] X. Yan, D. Song, and X. Li. Concept-based document readability in domain specific information retrieval. In *CIKM '06*, pages 540–549. ACM, 2006.

[27] J. Zhao, M.-Y. Kan, and Y. L. Theng. Math information retrieval: user requirements and prototype implementation. In *JCDL*, pages 187–196, 2008.