Explorations in Automatic Book Summarization

Rada Mihalcea and **Hakan Ceylan**
Department of Computer Science
University of North Texas
rada@cs.unt.edu, hakan@unt.edu

Abstract

Most of the text summarization research carried out to date has been concerned with the summarization of short documents (e.g., news stories, technical reports), and very little work if any has been done on the summarization of very long documents. In this paper, we try to address this gap and explore the problem of book summarization. We introduce a new data set specifically designed for the evaluation of systems for book summarization, and describe summarization techniques that explicitly account for the length of the documents.

1 Introduction

Books represent one of the oldest forms of written communication and have been used since thousands of years ago as a means to store and transmit information. Despite this fact, given that a large fraction of the electronic documents available online and elsewhere consist of short texts such as Web pages, news articles, scientific reports, and others, the focus of natural language processing techniques to date has been on the automation of methods targeting short documents. We are witnessing however a change: an increasingly larger number of books become available in electronic format, in projects such as Gutenberg (http://www.gutenberg.org), Google Book Search (http://books.google.com), or the Million Books project (http://www.archive.org/details/millionbooks). Similarly, a large number of the books published in recent years are often available – for purchase or through libraries – in electronic format. This means that the need for language processing techniques able to handle very large documents such as books is becoming increasingly important.

In this paper, we address the problem of *book summarization*. While there is a significant body of research that has been carried out on the task of text summarization, most of this work has been concerned with the summarization of *short* documents, with a particular focus on news stories. However, books are different in both length and genre, and consequently different summarization techniques are required. In fact, the straight-forward application of a current state-of-the-art summarization tool leads to poor results – a mere 0.348 F-measure compared to the baseline of 0.325 (see the following sections for details). This is not surprising since these systems were developed specifically for the summarization of short news documents.

The paper makes two contributions. First, we introduce a new data set specifically designed for the evaluation of book summaries. We describe the characteristics of a new benchmark consisting of books with manually constructed summaries, and we calculate and provide lower and upper performance bounds on this data set. Second, after briefly describing a summarization system that has been successfully used for the summarization of short documents, we show how techniques that take into account the length of the documents can be used to significantly improve the performance of this system.

2 Related Work

Automatic summarization has received a lot of attention from the natural language processing community, ever since the early approaches to automatic abstraction that laid the foundations of the current text summarization techniques (Luhn, 1958; Edmunson, 1969). The literature typically distinguishes between *extraction*, concerned with the identification of the information that is important in the input text; and *abstraction*, which involves a generation step to add fluency to a previously compressed text (Hovy and Lin, 1997). Most of the efforts to date have been concentrated on the extraction step, which is perhaps the most critical component of a successful summarization algorithm, and this is the focus of our current work as well.

To our knowledge, no research work to date was specifically concerned with the automatic summarization of *books*. There is, however, a large and growing body of work concerned with the summarization of short documents, with evaluations typically focusing on news articles. In particular, a significant number of summarization systems have been proposed during the recent Document Understanding Conference exercises (DUC) – annual evaluations that usually draw the participation of 20–30 teams every year.

There are two main trends that can be identified in the summarization literature: *supervised* systems, that rely on machine learning algorithms trained on pre-existing document-summary pairs, and *unsupervised* techniques, based on properties and heuristics derived from the text.

Among the unsupervised techniques, typical summarization methods account for both the weight of the words in sentences, as well as the sentence position inside a document. These techniques have been successfully implemented in the centroid approach (Radev et al., 2004), which extends the idea of *tf.idf* weighting (Salton and Buckley, 1997) by introducing word centroids, as well as integrating other features such as position, first-sentence overlap and sentence length. More recently, graph-based methods that rely on sentence connectivity have also been found successful, using algorithms such as node degree (Salton et al., 1997) or eigenvector centrality (Mihalcea and Tarau, 2004; Erkan and Radev, 2004; Wolf and Gibson, 2004).

In addition to unsupervised methods, supervised machine learning techniques have also been used with considerable success. Assuming the availability of a collection of documents and their corresponding manually constructed summaries, these methods attempt to identify the key properties of a good summary, such as the presence of named entities, positional scores, or the location of key phrases. Such supervised techniques have been successfully used in the systems proposed by e.g. (Teufel and Moens, 1997; Hirao et al., 2002; Zhou and Hovy, 2003; D'Avanzo and Magnini, 2005).

In addition to short news documents, which have been the focus of most of the summarization systems proposed to date, work has been also carried out on the summarization of other types of documents. This includes systems addressing the summarization of e-mail threads (Wan and McKeown, 2004), online discussions (Zhou and Hovy, 2005), spoken dialogue (Galley, 2006), product reviews (Hu and Liu, 2004), movie reviews (Zhuang et al., 2006), or short literary fiction stories (Kazantseva and Szpakowicz, 2006). As mentioned before, we are not aware of any work addressing the task of automatic book summarization.

3 A Data Set for the Evaluation of Book Summarization

A first challenge we encountered when we started working on the task of book summarization was the lack of a suitable data set, designed specifically for the evaluation of summaries of long documents. Unlike the summarization of short documents, which benefits from the data sets made available through the annual DUC evaluations, we are not aware of any publicly available data sets that can be used for the evaluation of methods for book summarization.

The lack of such data sets is perhaps not surprising since even for humans the summarization of books is more difficult and time consuming than the summarization of short news documents. Moreover, books are often available in printed format and are typically protected by copyright laws that do not allow their reproduction in electronic format, which consequently prohibits their public distribution.

We constructed a data set starting from the observation that several English and literature courses make use of books that are sometimes also available in the form of abstracts – meant to ease the access of students to the content of the books. In particular, we have identified two main publishers that make summaries available online for books studied in the U.S. high-school and college systems: Grade Saver (http://www.gradesaver.com) and Cliff's Notes (http://www.cliffsnotes.com/). Fortunately, many of these books are classics that are already in the public domain, and thus for most of them we were able to find the online electronic version of the books on sites such as Gutenberg or Online Literature (http://www.online-literature.com).

For instance, the following is an example drawn from Cliff's Notes summary of *Bleak House* by Charles Dickens.

> On a raw November afternoon, London is enshrouded in heavy fog made harsher by chimney smoke. The fog seems thickest in the vicinity of the High Court of Chancery. The court, now in session, is hearing an aspect of the case of Jarndyce and Jarndyce. A ”little mad old woman” is, as always, one of the spectators. Two ruined men, one a ”sallow prisoner,” the other a man from Shropshire, appear before the court – to no avail. Toward the end of the sitting, the Lord High Chancellor announces that in the morning he will meet with ”the two young people” and decide about making them wards of their cousin....

Starting with the set of books that had a summary available from Cliff's Notes, we removed all the books that did not have an online version, and further eliminated those that did not have a summary available from Grade Saver. This left us with a "gold standard" data set of 50 books, each of them with two manually created summaries.

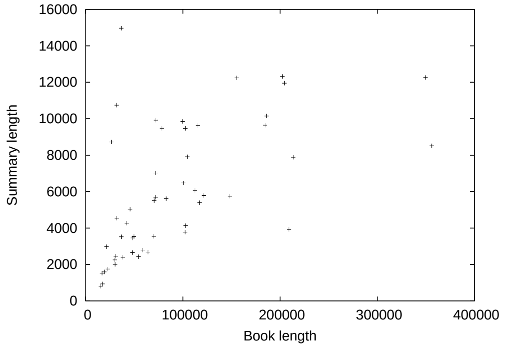

Figure 1: Summary and book lengths for 50 books

The books in this collection have an average length of 92,000 words, with summaries with an average length of 6,500 words (Cliff's Notes) and 7,500 words (Grade Saver). Figure 1 plots the length of the summaries (averaged over the two manual summaries) with respect to the length of the books. As seen in the plot, most of the books have a length of 50,000-150,000 words, with a summary of 2,000–6,000 words, corresponding to a compression rate of about 5-15%. There are also a few very long books, with more than 150,000 words, for which the summaries tend to become correspondingly longer.

3.1 Evaluation Metrics

For the evaluation, we use the ROUGE evaluation toolkit. ROUGE is a method based on Ngram statistics, found to be highly correlated with human evaluations (Lin and Hovy, 2003).[1] Throughout the paper, the evaluations are reported using the ROUGE-1 setting, which seeks unigram matches between the generated and the reference summaries, and which was found to have high correlation with human judgments at a 95% confidence level. Additionally, the final system is also evaluated using the ROUGE-2 (bigram matches) and ROUGE-SU4 (non-contiguous bigrams) settings, which have been frequently used in the DUC evaluations.

In most of the previous summarization evaluations, the data sets were constructed specifically for the purpose of enabling system evaluations, and thus the length of the reference and the generated summaries was established prior to building the data set and prior to the evaluations. For instance, some of the previous DUC evaluations provided reference summaries of 100-word each, and required the participating systems to generate summaries of the same length.

However, in our case we have to deal with pre-existing summaries, with large summary-length variations across the 50 books and across the two reference summaries. To address this problem, we decided to keep one manual summary as the main reference (Grade Saver), and use the other summary (Cliff's Notes) as a way to decide on the length of the generated summaries. This means that for a given book, the Cliff's Notes summary and all the

[1] ROUGE is available at http://haydn.isi.edu/ROUGE/

automatically generated summaries have the same length, and they are all evaluated against the (possibly with a different length) Grade Saver summary. This way, we can also calculate an upper bound by comparing the two manual summaries against each other, and at the same time ensure a fair comparison between the automatically generated summaries and this upper bound.[2]

3.2 Lower and Upper Bounds

To determine the difficulty of the task on the 50 book data set, we calculate and report lower and upper bounds. The lower bound is determined by using a baseline summary constructed by including the first sentences in the book (also known in the literature as the *lead* baseline).[3] As mentioned in the previous section, all the generated summaries – including this baseline – have a length equal to the Cliff's Notes manual summary. The upper bound is calculated by evaluating Cliff's Notes manual summary against the reference Grade Saver summary. Table 1 shows the precision (P), recall (R), and F-measure (F) for these lower and upper bounds, calculated as average across the 50 books.

	P	R	F
Lower bound (lead baseline)	0.380	0.284	0.325
Upper bound (manual summary)	0.569	0.493	0.528

Table 1: Lower and upper bounds for the book summarization task, calculated on the 50 book data set

An automatic system evaluated on this data set is therefore expected to have an F-measure higher than the lower bound of 0.325, and it is unlikely to exceed the upper bound of 0.528 obtained with a human-generated summary.

4 An Initial Summarization System

Our first book summarization experiment was done using a re-implementation of an existing state-of-the-art summarization system. We decided to use the centroid-based method implemented in the MEAD system (Radev et al., 2004), for three main reasons. First, MEAD was shown to lead to good performance in several DUC evaluations, e.g., (Radev et al., 2003; Li et al., 2005). Second, it is an unsupervised method which, unlike supervised approaches, does not require training data (not available in our case). Finally, the centroid-based techniques implemented in MEAD can be optimized and made very efficient, which is an important aspect in the summarization of very long documents such as books.

The latest version of MEAD[4] uses features, classifiers and re-rankers to determine the sentences to include in the summary. The default features are centroid, position and sentence length. The centroid value of a sentence is the sum of the centroid values of the words in the sentence. The centroid value of a word is calculated by multiplying the term frequency (*tf*) of a word by the word's inverse document frequency (*idf*) obtained from the Topic Detection and Tracking (TDT) corpus. The *tf* of a word is calculated by dividing the frequency of a word in a document cluster by the number of documents in the cluster. The positional value P_i of a sentence is calculated using the formula (Radev et al., 2004):

$$P_i = \frac{n - i + 1}{n} * C_{max} \qquad (1)$$

where n represents the number of sentences in the document, i represents the position of the sentence inside the text, and C_{max} is the score of the sentence that has the maximum centroid value.

The summarizer combines these features to give a score to each sentence. The default setting consists of a linear combination of features that assigns equal weights to the centroid and the positional values, and only scores sentences that have more than nine words. After the sentences are scored, the re-rankers are used to modify the scores of a sentence depending on its relation with other sentences. The default re-ranker implemented in MEAD first ranks the sentences by their scores in descending order and iteratively adds the top ranked sentence if the sentence is not *too similar* to the already added sentences. This similarity is computed as a cosine similarity and by default the sentences that exhibit a cosine similarity higher than 0.7 are not added to the

[2] An alternative solution would be to determine the length of the generated summaries using a predefined compression rate (e.g., 10%). However, this again implies great variations across the lengths of the generated versus the manual summaries, which can result in large and difficult to interpret variations across the ROUGE scores.

[3] A second baseline that accounts for text segments is also calculated and reported in section 6.

[4] MEAD 3.11, http://www.summarization.com/mead/

summary. Note that although the MEAD distribution also includes an optional feature calculated using the LexRank graph-based algorithm (Erkan and Radev, 2004), this feature could not be used since it takes days to compute for very long documents such as ours, and thus its application was not tractable.

Although the MEAD system is publicly available for download, in order to be able to make continuous modifications easily and efficiently to the system as we develop new methods, we decided to write our own implementation. Our implementation differs from the original one in certain aspects. First, we determine document frequency counts using the British National Corpus (BNC) rather than the TDT corpus. Second, we normalize the sentence scores by dividing the score of a sentence by the length of the sentence, and instead we eliminate the sentence length feature used by MEAD. Note also that we do not take stop words into account when calculating the length of a sentence. Finally, since we are not doing *multi*-document summarization, we do not use a re-ranker in our implementation.

	P	R	F
MEAD (original download)	0.423	0.296	0.348
MEAD (our implementation)	0.435	0.323	0.369

Table 2: Summarization results using the MEAD system

Table 2 shows the results obtained on the 50 book data set using the original MEAD implementation, as well as our implementation. Although the performance of this system is clearly better than the baseline (see Table 1), it is nonetheless far below the upper bound. In the following section, we explore techniques for improving the quality of the generated summaries by accounting for the length of the documents.

5 Techniques for Book Summarization

We decided to make several changes to our initial system, in order to account for the specifics of the data set we work with. In particular, our data set consists of *very large* documents, and correspondingly the summarization of such documents requires techniques that account for their length.

5.1 Sentence Position In Very Large Documents

The general belief in the text summarization literature (Edmunson, 1969; Mani, 2001) is that the position of sentences in a text represents one of the most important sources of information for a summarization system. In fact, a summary constructed using the lead sentences was often found to be a competitive baseline, with only few systems exceeding this baseline during the recent DUC summarization evaluations.

Although the position of sentences in a document seems like a pertinent heuristic for the summarization of short documents, and in particular for the newswire genre as used in the DUC evaluations, our hypothesis is that this heuristic may not hold for the summarization of very long documents such as books. The style and topic may change several times throughout a book, and thus the leading sentences will not necessarily overlap with the essence of the document.

To test this hypothesis, we modified our initial system so that it does not account for the position of the sentences inside a document, but it only accounts for the weight of the constituent words. Correspondingly, the score of a sentence is determined only as a function of the word centroids, and excludes the positional score. Table 3 shows the average ROUGE scores obtained using the summarization system with and without the position scores.

	P	R	F
With positional scores	0.435	0.323	0.369
Without positional scores	0.459	0.329	0.383

Table 3: Summarization results with and without positional scores

As suspected, removing the position scores leads to a better overall performance, with an increase observed in both the precision and the recall of the system. Although the position in a document is a heuristic that helps the summarization of news stories and other short documents, it appears that the sentences located toward the beginning of a book are not necessarily useful for building the summary of a book.

5.2 Text Segmentation

A major difference between short and long documents stands in the frequent topic shifts typically observed in the later. While short stories are usually concerned with one topic at a time, long documents such as books often cover more than one topic. Thus, the intuition is that a summary should include content covering the important aspects of *all* the topics in the document, as opposed to only generic aspects relevant to the document as a whole. A system for the summarization of *long* documents should therefore extract key concepts from all the topics in the document, and this task is better performed when the topic boundaries are known prior to the summarization step.

To accomplish this, we augment our system with a text segmentation module that attempts to determine the topic shifts, and correspondingly splits the document into smaller segments. Note that although chapter boundaries are available in some of the books in our data set, this is not always the case as there are also books for which the chapters are not explicitly identified. To ensure an uniform treatment of the entire data set, we decided not to use chapter boundaries, and instead apply an automatic text segmentation algorithm.

While several text segmentation systems have been proposed to date, we decided to use a graph-based segmentation algorithm using normalized-cuts (Malioutov and Barzilay, 2006), shown to exceed the performance of alternative segmentation methods. Briefly, the segmentation algorithm starts by modeling the text as a graph, where sentences are represented as nodes in the graph, and inter-sentential similarities are used to draw weighted edges. The similarity between sentences is calculated using cosine similarity, with a smoothing factor that adds the counts of the words in the neighbor sentences. Words are weighted using an adaptation of the *tf.idf* metric, where a document is uniformly split into chunks that are used for the *tf.idf* computation. There are two parameters that have to be set in this algorithm: (1) the length in words of the blocks approximating sentences; and (2) the cut-off value for drawing edges between nodes. Since the method was originally developed for spoken lecture segmentation, we were not able to use the same parameters as suggested in (Malioutov and Barzilay, 2006). Instead, we used a development set of three books, and determined the optimal sentence word-length as 20 and the optimal cut-off value as 25, and these are the values used throughout our experiments.

Once the text is divided into segments, we generate a separate summary for each segment, and consequently create a final summary by collecting sentences from the individual segment summaries in a round-robin fashion. That is, starting with the ranked list of sentences generated by the summarization algorithm for each segment, we pick one sentence at a time from each segment summary until we reach the desired book-summary length.

A useful property of the normalized-cut segmentation algorithm is that one can decide apriori the number of segments to be generated, and so we can evaluate the summarization algorithm for different segmentation granularities. Figure 2 shows the average ROUGE-1 F-measure score obtained for summaries generated using one to 50 segments.

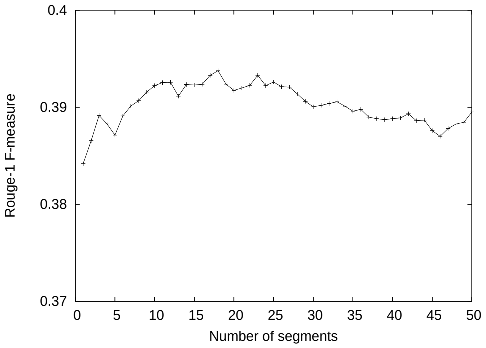

Figure 2: Summarization results for different segmentation granularities.

As seen in the figure, segmenting the text helps the summarization process. The average ROUGE-1 F-measure score raises to more than 0.39 F-measure for increasingly larger number of segments, with a plateau reached at approximately 15–25 segments, followed by a decrease when more than 30 segments are used.

In all the following evaluations, we segment each book into a constant number of 15 segments; in future work, we plan to consider more sophisticated methods for finding the optimal number of segments individually for each book.

5.3 Modified Term Weighting

An interesting characteristic of documents with topic shifts is that words do not have an uniform distribution across the entire document. Instead, their distribution can vary with the topic, and thus the weight of the words should change accordingly.

To account for the distribution of the words inside the entire book, as well as inside the individual topics (segments), we devised a weighting scheme that accounts for four factors: the *segment term frequency (stf)*, calculated as the number of occurrences of a word inside a segment; the *book term frequency (tf)*, determined as the number of occurrences of a word inside a book; the *inverse segment frequency (isf)*, measured as the inverse of the number of segments containing the word; and finally, the *inverse document frequency (idf)*, which takes into account the distribution of a word in a large external corpus (as before, we use the BNC corpus). A word weight is consequently determined by multiplying the book term frequency with the segment term frequency, and the result is then multiplied with the inverse segment frequency and the inverse document frequency. We refer to this weighting scheme as *tf.stf.idf.isf*.

Using this weighting scheme, we prevent a word from having the same score across the entire book, and instead we give a higher weight to its occurrences in segments where the word has a high frequency. For instance, the word *doctor* occurs 30 times in one of the books in our data set, which leads to a constant *tf.idf* score of 36.76 across the entire book. Observing that from these 30 occurrences, 19 appear in just one segment, the *tf.stf.idf.isf* weighting scheme will lead to a weight of 698.49 for that segment, much higher than e.g. the weight of 36 calculated for other segments that have only a few occurrences of this word.

	P	R	F
tf.idf weighting	0.463	0.339	0.391
tf.stf.idf.isf weighting	0.464	0.349	0.398

Table 4: Summarization results using a weighting scheme accounting for the distribution of words inside and across segments

Table 4 shows the summarization results obtained for the new weighting scheme (recall that all the results are calculated for a text segmentation into 15 segments).

5.4 Combining Summarization Methods

The next improvement we made was to bring an additional source of knowledge into the system, by combining the summarization provided by our current system with the summarization obtained from a different method.

We implemented a variation of a centrality graph-based algorithm for unsupervised summarization, which was successfully used in the past for the summarization of short documents. Very briefly, the TextRank system (Mihalcea and Tarau, 2004) – similar in spirit with the concurrently proposed LexRank method (Erkan and Radev, 2004) – works by building a graph representation of the text, where sentences are represented as nodes, and weighted edges are drawn using inter-sentential word overlap. An eigenvector centrality algorithm is then applied on the graph (e.g., PageRank), leading to a ranking over the sentences in the document. An impediment we encountered was the size of the graphs, which become intractably large and dense for very large documents such as books. In our implementation we decided to use a cut-off value for drawing edges between nodes, and consequently removed all the edges between nodes that are farther apart than a given threshold. We use a threshold value of 75, found to work best using the same development set of three books used before.

	P	R	F
Our system	0.464	0.349	0.398
TextRank	0.449	0.356	0.397
COMBINED	0.464	0.363	0.407

Table 5: Summarization results for individual and combined summarization algorithms

Using the same segmentation as before (15 segments), the TextRank method by itself did not lead to improvements over our current centroid-based system. Instead, since we noticed that the summaries generated with our system and with TextRank covered different sentences, we implemented a method that combines the top ranked sentences from the two methods. Specifically, the combination method picks one sentence at a time from the summary generated by our system for each segment, followed by one sentence selected from the summary generated by the TextRank method, and so on. The combination method also specifically avoids redundancy. Table 5 shows the results obtained with our current centroid-based system the TextRank method, as well as the combined method.

5.5 Segment Ranking

In the current system, all the segments identified in a book have equal weight. However, this might not always be the case, as there are sometimes topics inside the book that have higher importance, and which consequently should be more heavily represented in the generated summaries.

To account for this intuition, we implemented a segment ranking method that assigns to each segment a score reflecting its importance inside the book. The ranking is performed with a method similar to TextRank, using a random-walk model over a graph representing segments and segment similarities. The resulting segment scores are multiplied with the sentence scores obtained from the combined method described before, normalized over each segment, resulting in a new set of scores. The top ranked sentences over the entire book are then selected for inclusion in the summary. Table 6 shows the results obtained by using segment ranking.

	P	R	F
COMBINED	0.464	0.363	0.407
COMBINED + Segment Ranking	0.472	0.366	0.412

Table 6: Summarization results using segment ranking

6 Discussion

In addition to the ROUGE-1 metric, the quality of the summaries generated with our final summarization system was also evaluated using the ROUGE-2 and the ROUGE-SU4 metrics, which are frequently used in the DUC evaluations. Table 7 shows the figures obtained with ROUGE-1, ROUGE-2 and ROUGE-SU4 for our final system, for the original MEAD download, as well as for the lower and upper bounds. The table also shows an additional baseline determined by selecting the first sentences in each segment, using the segmentation into 15 segments as determined before. As it can be seen from the F-measure confidence intervals also shown in the table, the improvements obtained by our system with respect to both baselines and with respect to the MEAD system are statistically significant (as the confidence intervals do not overlap).

	P	R	F
	ROUGE-1		
Lower bound	0.380	0.284	0.325 [0.306,0.343]
Segment baseline	0.402	0.301	0.344 [0.328,0.366]
MEAD	0.423	0.296	0.348 [0.329,0.368]
Our system	**0.472**	**0.366**	**0.412 [0.394,0.428]**
Upper bound	0.569	0.493	0.528 [0.507,0.548]
	ROUGE-2		
Lower bound	0.035	0.027	0.031 [0.027,0.035]
Segment baseline	0.040	0.031	0.035 [0.031,0.038]
MEAD	0.039	0.029	0.033 [0.028,0.037]
Our system	**0.069**	**0.054**	**0.061 [0.055,0.067]**
Upper bound	0.112	0.097	0.104 [0.096,0.111]
	ROUGE-SU4		
Lower bound	0.096	0.073	0.083 [0.076,0.090]
Segment baseline	0.102	0.079	0.089 [0.082,0.093]
MEAD	0.106	0.076	0.088 [0.081,0.095]
Our system	**0.148**	**0.115**	**0.129 [0.121,0.138]**
Upper bound	0.210	0.182	0.195 [0.183,0.206]

Table 7: Evaluation of our final book summarization system using different ROUGE metrics. The table also shows: the lower bound (first sentences in the book); the segment baseline (first sentences in each segment); MEAD (original system download); the upper bound (manual summary). Confidence intervals for F-measure are also included.

Additionally, to determine the robustness of the results with respect to the number of reference summaries, we ran a separate evaluation where both the Grade Saver and the Cliff's Notes summaries were used as reference. As before, the length of the generated summaries was determined based on the Cliff's Notes summary. The F-measure figures obtained in this case using our summarization system were 0.402, 0.057 and 0.127 using ROUGE-1, ROUGE-2 and ROUGE-SU4 respectively. The F-measure figures calculated for the baseline using the first sentences in each segment were 0.340, 0.033 and 0.085. These figures are very close to those listed in Table 7 where only one summary was used as a reference, suggesting that the use of more than one reference summary does not influence the results.

Regardless of the evaluation metric used, the performance of our book summarization system is significantly higher than the one of an existing summarization system that has been designed for the summarization of short documents (MEAD). In fact, if we account for the upper bound of 0.528, the relative error rate reduction for the ROUGE-1 F-measure score obtained by our system with respect to MEAD is a significant 34.44%.

The performance of our system is mainly due to features that account for the length of the document: exclusion of positional scores, text segmentation and segment ranking, and a segment-based weighting scheme. An additional improvement is obtained by combining two different summarization methods. It is also worth noting that our system is efficient, taking about 200 seconds to apply the segmentation algorithm, plus an additional 65 seconds to generate the summary of one book.[5]

To assess the usefulness of our system with respect to the length of the documents, we analyzed the individual results obtained for books of different sizes. Averaging the results obtained for the shorter books in our collection, i.e., 17 books with a length between 20,000 and 50,000 words, the lead baseline gives a ROUGE-1 F-measure score of 0.337, our system leads to 0.378, and the upper bound is measured at 0.498, indicating a relative error rate reduction of 25.46% obtained by our system with respect to the lead baseline (accounting for the maximum achievable score given by the upper bound). Instead, when we consider only the books with a length over 100,000 words (16 books in our data set fall under this category), the lead baseline is determined as 0.347, our system leads to 0.418, and the upper bound is calculated as 0.552, which results in a higher 34.64% relative error rate reduction. This suggests that our system is even more effective for longer books, due perhaps to the features that specifically take into account the length of the books.

There are also cases where our system does not improve over the baseline. For instance, for the summarization of *Candide* by François Voltaire, our system achieves a ROUGE-1 F-measure of 0.361, which is slightly worse than the lead baseline of 0.368. In other cases however, the performance of our system comes close to the upper bound, as it is the case with the summarization of *The House of the Seven Gables* by Nathaniel Hawthorne, which has a lead baseline of 0.296, an upper bound of 0.457, and our system obtains 0.404. This indicates that a possible avenue for future research is to account for the characteristics of a book, and devise summarization methods that can adapt to the specifics of a given book such as length, genre, and others.

7 Conclusions

Although there is a significant body of work that has been carried out on the task of text summarization, most of the research to date has been concerned with the summarization of *short* documents. In this paper, we tried to address this gap and tackled the problem of book summarization.

We believe this paper made two important contributions. First, it introduced a new summarization benchmark, specifically targeting the evaluation of systems for book summarization.[6] Second, it showed that systems developed for the summarization of short documents do not fare well when applied to very long documents such as books, and instead a better performance can be achieved with a system that accounts for the length of the documents. In particular, the book summarization system we developed was found to lead to more than 30% relative error rate reduction with respect to an existing state-of-the-art summarization tool.

Given the increasingly large number of books available in electronic format, and correspondingly the growing need for tools for book summarization, we believe that the topic of automatic book summarization will become increasingly important. We hope that this paper will encourage and facilitate the development of an active line of research concerned with book summarization.

Acknowledgments

The authors are grateful to the Language and Information Technologies research group at the University of North Texas for useful discussions and feedback, and in particular to Carmen Banea for suggesting Cliff's Notes as a source of book summaries. This work was supported in part by a research grant from Google Inc. and by a grant from the Texas Advanced Research Program (#003594).

[5] Running times measured on a Pentium IV 3GHz, 2GB RAM.

[6] The data set is publicly available and can be downloaded from http://lit.csci.unt.edu/index.php/Downloads

References

E. D'Avanzo and B. Magnini. 2005. A keyphrase-based approach to summarization: The Lake system at DUC 2005. In *Proceedings of the Document Understanding Conference (DUC 2005)*.

H.P. Edmunson. 1969. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285.

G. Erkan and D. Radev. 2004. Lexpagerank: Prestige in multi-document text summarization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain, July.

M. Galley. 2006. Automatic summarization of conversational multi-party speech. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI 2006), AAAI/SIGART Doctoral Consortium*, Boston.

T. Hirao, Y. Sasaki, H. Isozaki, and E. Maeda. 2002. NTT's text summarization system for DUC-2002. In *Proceedings of the Document Understanding Conference 2002 (DUC 2002)*.

E. Hovy and C. Lin, 1997. *Automated text summarization in SUMMARIST*. Cambridge Univ. Press.

M. Hu and B. Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, Seattle, Washington.

A. Kazantseva and S. Szpakowicz. 2006. Challenges in evaluating summaries of short stories. In *Proceedings of the Workshop on Task-Focused Summarization and Question Answering*, Sydney, Australia.

W. Li, W. Li, B. Li, Q. Chen, and M. Wu. 2005. The Hong Kong Polytechnic University at DUC 2005. In *Proceedings of the Document Understanding Conference (DUC 2005)*, Vancouver, Canada.

C.Y. Lin and E.H. Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.

H. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

I. Malioutov and R. Barzilay. 2006. Minimum cut model for spoken lecture segmentation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 9–16.

I. Mani. 2001. *Automatic Summarization*. John Benjamins.

R. Mihalcea and P. Tarau. 2004. TextRank – bringing order into texts. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004)*, Barcelona, Spain.

D. Radev, J. Otterbacher, H. Qi, and D. Tam. 2003. MEAD ReDUCs: Michigan at DUC 2003. In *Proceedings of the Document Understanding Conference (DUC 2003)*.

D. Radev, H. Jing, M. Stys, and D. Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing and Management*, 40.

G. Salton and C. Buckley. 1997. Term weighting approaches in automatic text retrieval. In *Readings in Information Retrieval*. Morgan Kaufmann Publishers, San Francisco, CA.

G. Salton, A. Singhal, M. Mitra, and C. Buckley. 1997. Automatic text structuring and summarization. *Information Processing and Management*, 2(32).

S. Teufel and M. Moens. 1997. Sentence extraction as a classification task. In *ACL/EACL workshop on intelligent and scalable text summarization*, Madrid, Spain.

S. Wan and K. McKeown. 2004. Generating overview summaries of ongoing email thread discussions. In *Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland.

F. Wolf and E. Gibson. 2004. Paragraph-, word-, and coherence-based approaches to sentence ranking: A comparison of algorithm and human performance. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL 2004)*, Barcelona, Spain, July.

L. Zhou and E. Hovy. 2003. A Web-trained extraction summarization system. In *Proceedings of Human Language Technology Conference (HLT-NAACL 2003)*, Edmonton, Canada, May.

L. Zhou and E. Hovy. 2005. Digesting virtual ”geek” culture: The summarization of technical internet relay chats. In *Proceedings of Association for Computational Linguistics (ACL 2005)*, Ann Arbor.

L. Zhuang, F. Jing, and X.Y. Zhu. 2006. Movie review mining and summarization. In *Proceedings of the ACM international conference on Information and knowledge management (CIKM 2006)*, Arlington, Virginia.