Information-Content Based Sentence Extraction for Text Summarization

Daniel Mallett[1] James Elding[2] Mario A. Nascimento[1]

[1]Department of Computing Science, University of Alberta, Canada
{mallett, mn}@cs.ualberta.ca
[2]Workers' Compensation Board, Alberta, Canada
james.elding@wcb.ab.ca

Abstract

This paper proposes the FULL-COVERAGE *summarizer: an efficient, information retrieval oriented method to extract non-redundant sentences from text for summarization purposes. Our method leverages existing Information Retrieval technology by extracting key-sentences on the premise that the relevance of a sentence is proportional to its similarity to the whole document. We show that our method can produce sentence-based summaries that are up to 78% smaller than the original text with only 3% loss in retrieval performance.*

1 Introduction

Text summarization is the process of condensing a source text while preserving its information content and maintaining readability. The main (and large) difference between automatic and human-based text summarization is that humans can capture and convey subtle themes that permeate documents, whereas automatic approaches have a large difficulty to do the same. Nonetheless, as the amount of information available in electronic format continues to grow, research into automatic text summarization has taken on renewed interest. Indeed, A whole conference series, the Document Understanding Conferences (DUC)[1] has been devoted to the topic in the recent past.

We model the summarization problem as follows:

> *Given a document D, composed of a set of N sentences $S = \{S_1, S_2, ..., S_N\}$, and a percentage p, select a subset of sentences $S' \subseteq S$ such that: (1) $|S'| \leq p \times |S|$ and (2) using S' as a representative of D is as effective as using S in terms of retrieval performance.*

To address this problem, most existing text summarizers, e.g. [9, 13, 15], use the framework given in Figure 1.

```
INPUT  : Full text for document $D$
OUTPUT: Summary text of document $D$
  1: Parse the document into sentences.
  2: Determine the saliency (rank) of each sentence.
  3: Generate a summary using a subset of the ranked
     sentences (usually a few top ones).
```

Figure 1: Text Summarization Framework

Summaries are generally evaluated along two orthogonal axes: information content (our focus) and readability. Our goal is to find a set of non-redundant sentences faithful to the information content of the original text (steps 1 and 2 in Figure 1) from which one could generate high-quality human readable summaries (step 3 in Figure 1).

Most existing summarization techniques use some variant of the well known $tf * idf$ model [2]. Our algorithm extracts sentences that "fully covers" the concept-space of a document by iteratively measuring the similarity of each sentence to the whole document and striking-out words that have already been covered. As our experiments corroborate, the FULL-COVERAGE approach is a simple yet efficient alternative for steps 1 and 2 of the framework in Figure 1.

One of the great challenges of text summarization is summary evaluation [14]. In order to evaluate how well our sentence-extractor performs, we propose an extrinsic retrieval-oriented evaluation that, unlike most previous work, does not rely on human assessors. Our experimental methodology aims at measuring how well a summary captures the salient information content of a document. We experiment with data from SMART's TIME Magazine Collection[2] as well as the TREC[3] documents used for 2002's edition of DUC. As we will discuss, the results from our eval-

[1] http://duc.nist.gov

[2] ftp://ftp.cs.cornell.edu/pub/smart/time/

[3] http://trec.nist.gov

uation are promising. A summarized version of the TIME Magazine collection, 40% the size of the original text, loses only about 5% in terms of retrieval precision. The DUC results are even more interesting; our FULL-COVERAGE approach ranks highly against other DUC competitors, producing summaries 22% the size of the original texts with only a 3% loss in retrieval performance.

This paper is structured as follows. In Section 2 we review previous research related to the problem of text summarization and summary evaluation. Section 3 presents our FULL-COVERAGE key-sentence extraction method. Section 4 provides experiments comparing our method to ten other summarization approaches. Finally, Section 5 concludes the paper.

2 Related Work

Important seminal papers and recent advance papers on text summarization can be found in Mani and Maybury's book on text summarization [9]. The next section outlines recent advances in text summarization relevant to our FULL-COVERAGE approach, followed by a section discussing related evaluation methodologies.

2.1 Summarization Techniques

Text summarization by extraction can employ various levels of granularity, e.g., keyword, sentence, or paragraph. Most research concentrates on sentence-extraction because the readability of a list of keywords is typically low while paragraphs are unlikely to cover the information content of a document given summary space constraints.

MEAD [13], a state of the art sentence-extractor and a top performer at DUC, aims to extracts sentences central to the overall topic of a document. The system employs (1) a centroid score representing the centrality of a sentence to the overall document, (2) a position score which is inversely proportional to the position of a sentence in the document, and (3) an overlap-with-first score which is the inner product of the $tf * idf$ with the first sentence of the document. MEAD attempts to reduce summary redundancy by eliminating sentences above a similarity threshold parameter. As we will see, our proposal is simpler than MEAD and consistently outperformed it in the experiments we carried.

Other approaches for sentence extraction include NLP methods [1, 3] and machine-learning techniques [11, 16]. These approaches tend to be computationally expensive and genre-dependent even though they are typically based on the more general $tf * idf$ framework. Work on generative algorithms includes sentence compression [6], sentence fusion [5], and sentence modification[10]. We envision our FULL-COVERAGE approach as providing the input extracted sentences into these type of generative algorithms.

Maximal Marginal Relevance (MMR) is a technique explicitly concerned with reducing redundancy [4]. MMR reranks retrieval query results (relevant document lists) based on a document's dissimilarity to other relevant documents. The method is extended to summarize single-documents by re-ranking salient sentences instead of documents. Our redundancy reducing mechanism is strikingly different than that approach.

2.2 Summary Evaluation

Summaries can be evaluated using intrinsic or extrinsic measures [14]. While intrinsic methods attempt to measure summary quality using human evaluation thereof, extrinsic methods measure the same through a task-based performance measure such the information retrieval-oriented task. Typically, the former is used, e.g., [1, 13], [15] being a notable exception of the latter.

We propose an extrinsic IR evaluation that measures the information content of a summary with respect to the original document. We posit that our evaluation measures the ability of the summary to retain the information content of the original text, i.e., if the original text is relevant to a certain set of queries then the summary will be as well.

The TIPSTER/SUMMAC and NTCIR conferences have experimented with the retrieval-oriented task and use precision and recall in their evaluations. These works focus on the relevance assessment task, not the information retrieval task itself, and require considerable human involvement. Sakai et al [15] use a performance measure similar to our evaluation. A striking difference though is that they chose *a priori* which documents are relevant to their queries whereas we experiment with a standard collection of queries and relevant documents.

3 The FULL-COVERAGE Algorithm

The intuition behind our FULL-COVERAGE summarizer is to consider key-sentence extraction from an information retrieval perspective based on the premise that the relevance of a sentence is proportional to its similarity to the whole document. Like MMR [4], we aim to minimize summary redundancy, however our method is different from MMR in that we do not consider sentence dissimilarity, but instead focus on reducing summary redundancy by removing query terms that have already been covered by other salient sentences. If several sentences from a document share a set of common terms, i.e., all refer to the same concept, only a very small subset of those sentences, likely a single one, which covers the same concept space will be considered salient using our algorithm. Next we discuss how our proposed method fits in the framework shown in Figure 1.

The first phase of the FULL-COVERAGE algorithm (Figure 1 step 1) is to parse a document into sentences. We use a technique from [8] for this task. During this phase we also remove stop-words and apply the Porter stemming algorithm [12].

```
INPUT   : Document $D$
OUTPUT: FULL-COVERAGE of $D$
$Q \leftarrow D$
repeat
    $S^* \leftarrow S_i$ with max $sim(Q, S_i)$, $(1 \leq i \leq N)$
    $FC \leftarrow FC \cup S^*$
    Remove from $Q$ all terms occurring in $S^*$
until $Q$ not changed
Return $FC$
```

Figure 2: Algorithm to Compute the FULL-COVERAGE Set

The second step of the algorithm is to calculate FC – the subset of the sentences that cover the entire concept space of a document. Figure 2 materializes this phase of our algorithm in pseudo-code format. The running time of the algorithm is quadratic in the number of sentences in a document. The method for determining FC is to treat each individual sentence $S_i(i = 1, ..., N)$ of D as a document within the overall "collection" of D itself. The next step is to use the entire document as a query against each individual sentence, adding the highest ranked sentence to the full coverage set. Then any words that appear in the highest ranked sentence are stricken out from the query, so as to remove that concept from the document. The process repeats until no more words can be struck out from the query string.

$$\begin{array}{lccc} D: & \text{"A B C."} & = & S_1 \\ & \text{"E A."} & = & S_2 \\ & \text{"B A."} & = & S_3 \end{array}$$

1: $Q =$ "A B C E A B A" $\rightarrow S_1$

2: $Q =$ "$\cancel{A}$ $\cancel{B}$ $\cancel{C}$ E $\cancel{A}$ $\cancel{B}$ $\cancel{A}$"

3: $Q =$ "E" $\rightarrow S_2$

4: $Q =$ "$\cancel{E}$"

5: $Q =$ "" $\rightarrow$ stop

$$FC = \{S_1, S_2\}$$

Figure 3: Example of the FULL-COVERAGE Algorithm

An example of the algorithm over a simple document D containing three sentences is given in Figure 3. Initially, each sentence is obtained and weighted as if they were (sentence) documents S_1, S_2, and S_3 within the collection D. In step 1, by using all the keywords in D as the query string Q, the "document" S_1 will be returned as the most similar document, i.e., higher ranked sentence, and any occurrences of the words found in S_1 ("A", "B", and "C") are struck out from the query string Q (step 2). Querying again (step 3), S_2 is returned, and any occurrences of the words found in S_2 ("E" and "A") are struck out from Q (step 4). In step 5, the algorithm stops because Q is empty. The resulting full coverage set has covered all words from D.

Our experimentation employs only the standard $tf * idf$ weighting scheme as defined by the vector model [2] for implementing the $sim()$ function. The key advantage to using a single weighting scheme is that the algorithm in Figure 2 can be easily implemented on top of most (if not any) vector model based implementation such as that provided by MG [17]. A script to retrieve the documents to be summarized (which uses the sentence-breaker, stemming and stop-word removal tools) is used both by MG and our FULL-COVERAGE scripting application to retrieve documents already parsed by sentences. The FULL-COVERAGE code interacts with MG through MG's built-in indexing and querying interfaces to implement the FULL-COVERAGE algorithm.

Once the ranked FULL-COVERAGE set of sentences has been determined, the third step of Figure 1 is to actually generate and return a summary. Our aim is to achieve an acceptable trade-off between how much one can save in space overhead by using a summary while still retaining as much as possible the information content of the original document. Thus, in the next section we show experiments (1) with different collections, and (2) using summaries generated at different compression ratios $CR(p) = p \times |FC|$. Given a percentage p, a $CR(p)$ summary consists of the first n sentences from FC where $n \leq p * |FC|$ Another possibility, used in DUC evaluations, is to use a well-defined maximum number of terms per summary.

4 Experimental Results

In order to evaluate our FULL-COVERAGE approach we used the TIME Magazine collection from SMART and the TREC collection as used in DUC 2002. The collections contain documents covering very different domains. Recall that, unlike the work reviewed in Section 2.2, we do not rely on human evaluation. Our basic measure of performance is Precision-Recall (P-R) [2]. The F-measure, non-interpolated precision, and R-precision were also measured but are not reported here since they strongly correlate with interpolated P-R.

4.1 TIME Collection

The TIME collection consists of 423 documents with an average of 27.3 sentences/document, and 83 queries with an average of 3.9 relevant documents/query. In addition to our proposed method and the MEAD system [13][4], we have also

[4] Available on-line at `http://www.summarization.com`

used two other baseline techniques, namely, a random and a lead-based summarizer. The former simply selects unique sentences randomly, while the latter selects the first n sentences from S^j. Both approaches observe the size limit of the produced summary. The P-R curves obtained at various levels of compression are depicted in Figure 4 (curves for CR(1.0) and CR(0.8) are very similar to CR(0.6)). Recall that the compression ratio CR denotes how many sentences (percentage-wise) are to be actually used as the summary in terms of the number of sentences selected by the FULL-COVERAGE technique.

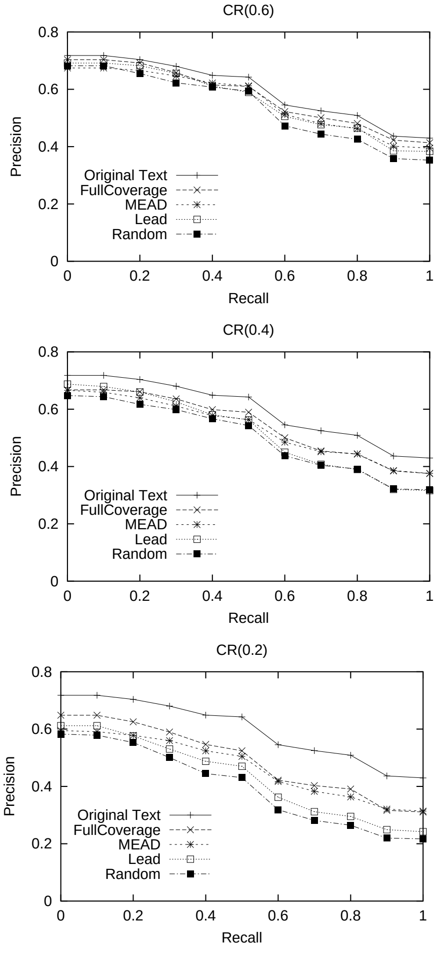

Figure 4: TIME collection interpolated P-R curves.

The FULL-COVERAGE technique proved to be effective. At nearly all levels of recall for all four CR values, the FULL-COVERAGE technique outperforms the MEAD, lead, and random summaries. Random summaries perform poorly overall and the lead summarizer performs adequately. In [7] it is reported that lead summaries for single documents can be effective. The lead summarizer performs poorly however, at lower levels of compression.

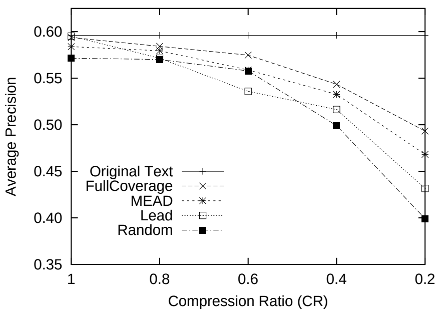

Figure 5: TIME collection average interpolated precision.

Figure 5 provides a comparison of all approaches based on the average interpolated precision over all levels of recall. FULL-COVERAGE has the slowest precision decrease as the compression ratio changes. Note that the compression rate in sentences is strongly correlated to the compression rate in terms of words. The FULL-COVERAGE technique clearly outperforms the other three approaches. These results are encouraging considering the relatively straightforward and computationally inexpensive nature of the approach. The FULL-COVERAGE technique, in particular at CR(0.4), offers a very good compromise in terms of summary length and information content.

4.2 DUC Collection

Every year, DUC evaluates competing research group's summarization systems on a set of summarization tasks. In DUC 2002, Task #1 was single-document summarization – the goal being to generate 100 word summaries (although this limit is not strictly enforced) of 533 documents from the TREC collection. Eight research groups (including MEAD) submitted summaries for all of the documents, which were evaluated by human evaluators. (More details on DUC, the participating groups and their summarization techniques are available on their website.) We computed an intersection of relevant documents for queries in TREC 9 with the documents in DUC 2002. Although the number of queries and relevant documents is less than ideal, the advantage of this evaluation is that we were able to compare a large number of systems to our own in a fully automatic mode.

The average precision results of the evaluation using the DUC 2002 summaries are provided in Figure 6. The "Baseline" corresponds to using the full text. The average length of each original document was 573 words. The results for the two lowest performing systems are omitted from Figure 6 for the sake of clarity. Generating 100 words summaries, the FULL-COVERAGE summarizer's average compression ratio in terms of words was 22% (a storage savings of 78%) – a substantial savings for only a 3% loss in average preci-

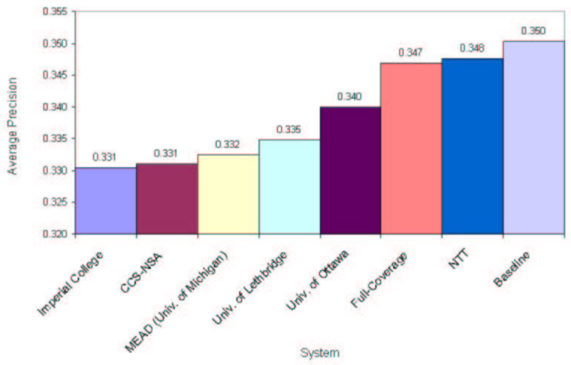

Figure 6: DUC 2002

sion. The results confirm that our FULL-COVERAGE summarizer performs well as compared to other state-of-the-art systems competing in DUC. We conclude that our FULL-COVERAGE algorithm is an effective method for extracting key-sentences from text.

5 Conclusions

We have presented a technique for extracting key-sentences from a document in order to use such sentences as a summary of the same. The technique can be implemented using an off-the-shelf information retrieval system, and has been experimentally shown to be effective at retaining the information content of a document.

To the best of our knowledge, the extrinsic retrieval task-based evaluation we have performed - without the interference of human evaluators - has not yet been performed. The results of our evaluation – that it is possible to achieve only a small loss (5% for the TIME collection and 3% for the DUC) in precision using a smaller (60% for the TIME collection and 78% for the DUC) text are very interesting. As well, experiments on the scale of our DUC experiments – comparing 9 systems along with the original text over 533 documents for each system – are also an original contribution, and raise interesting research questions. For instance, since most systems rely heavily on variants of $tf * idf$ weightings for extracting key-sentences and correspondingly perform equally well (all of the precision values in Figure 6 are within a small range), more emphasis should perhaps be put in investigating effective generative algorithms (i.e. step 3 of Figure 1), that can take the extracted key-sentences and present the information at a higher level of abstraction.

Acknowledgments

This research was partially supported by iCORE Alberta and NSERC Canada.

References

[1] C. Aone, M. E. Okurowski, J. Gorlinsky, and B. Larsen. A Scalable Summarization System Using Robust NLP. In *Proceedings of the Intelligent Scalable Text Summarization Workshop*, pages 66–73, 1997.

[2] R. A. Baeza-Yates and B. A. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press / Addison-Wesley, 1999.

[3] R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the Intelligent Scalable Text Summarization Workshop*, pages 10–17, 1997.

[4] J. Carbonell and J. Goldstein. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the ACM SIGIR conference on Research and Development in Information Retrieval*, pages 335–336, 1998.

[5] K. Han, Y. Song, and H. Rim. KU Text Summarization System for DUC 2003. In *Document Understanding Conference Draft Papers*, pages 118–121, 2003.

[6] C.-Y. Lin. Improving Summarization Performance by Sentence Compression - A Pilot Study. In *Proceedings of the International Workshop on Information Retrieval with Asian Language*, pages 1–8, 2003.

[7] C.-Y. Lin and E. Hovy. The Potential and Limitations of Automatic Sentence Extraction for Summarization. In *Text Summarization: Proceedings of the NLT-NAACL Workshop*, pages 73–80, 2003.

[8] D. Lin. LaTaT: Language and Text Analysis Tools. In *Proceedings of the Human Language Technology Conference*, pages 222–227, 2001.

[9] I. Mani and M. Maybury. *Advances in Automatic Text Summarization*. MIT Press, 1999.

[10] A. Nenkova, B. Schiffman, A. Schlaiker, S. Blair-Goldensohn, R. Barzilay, S. Sigelman, V. Hatzivassiloglou, and K. McKeown. Columbia at the Document Understanding Conference 2003. In *2003 Document Understanding Conference Draft Papers*, pages 71–78, 2003.

[11] C. Nobata and S. Sekine. Results of CRL/NYU System at DUC-2003 and an Experiment on Division of Document Sets. In *2003 Document Understanding Conference Draft Papers*, pages 79–85, 2003.

[12] M. Porter. An algorithm for suffix stripping. *Readings in Information Retrieval*, pages 130–137, 1980.

[13] D. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation and user studies. In *ANLP/NAACL Workshop on Automatic Summarization*, pages 21–29, 2000.

[14] D. R. Radev, E. Hovy, and K. McKeown. Introduction to the special issue on summarization. *Computational Linguistics*, 28(4):399–408, 2002.

[15] K. Sparck-Jones and T. Sakai. Generic Summaries for Indexing in IR. In *SIGIR Conference on Research and Development in Information Retrieval*, pages 190–198, 2001.

[16] S. Teufel and M. Moens. Sentence extraction as a classification task. In *ACL/EACL Workshop on Intelligent and Scalable Text Summarization*, 1997.

[17] I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. 1999.