Comparing Redundancy Removal Techniques for Multi–Document Summarisation

Eamonn Newman, William Doran, Nicola Stokes, Joe Carthy, John Dunnion
Intelligent Information Retrieval Group,
Department of Computer Science, University College Dublin, Ireland

Abstract. We describe an experiment to determine the quality of different similarity metrics, with regard to redundancy removal. The three metrics under examination are WordNet distance, Cosine Similarity (Vector-space model) and Latent Semantic Indexing. Our experiment was performed on a corpus drawn from the 2003 Document Understanding Conference (DUC)[1].

1 Introduction

Document summarisation systems automatically produce summaries of full–length documents. Summarisation systems can be broadly divided into two categories: the single–document summariser and the more complex multi–document system. In general, summarisation systems may be based on a number of different research areas, from simplistic methods such as word count [2], and term and document frequencies (*tf.idf*) [3] to linguistic approaches such as Discourse Structure [4].

Multi–document systems can be viewed as augmented single–document summarisers, i.e. they perform all the tasks of an SDS but must also cope with conflicts and contradictions[5], redundancy [5, 6], collation, sentence ordering, etc.

In extending a single–document summariser [7], the first step we have chosen is to implement *Redundancy Removal*. This will identify any information repetition in the source (input) texts, thus minimising any redundant or repetitive content in the final summary. This paper describes our experiment to determine the efficacy of certain similarity measures.

2 Text Summarisation

Automatic summarisation is the process by which the information in a source text is expressed in a more concise fashion, with a minimal loss of information. Summaries can be either *extractive* or *abstractive*. An extractive summary is built from sentences, phrases or words (depending on the degree of compression required) extracted from the source text. Abstractive summaries, on the other hand, involve an analysis of the text at a deeper level to determine the subject matter, and thereafter a reformulation of text to form a coherent passage, resulting in an accurate summary.

Topic Detection is a very important step in the summarisation process, as it identifies the most salient themes in a text. This can be done in a variety of ways, depending on the application. For example, in TREC Topic Detection and Tracking tasks [8], a Topic Detector must be able to find the topic of a text based on previously–seen articles, and to modify

its representation of the theme if necessary, where any changes in focus or fact may have occurred due to the temporal nature of the news domain.

Having identified the passages of text which are the most significant, the **redundancy removal** step attempts to reduce the size of this set without any reduction in information content. This is achieved by removing the sentences which duplicate information given in other sentences.

The methods we used to perform simple redundancy removal are given in Section 4. Methods under consideration for implementation in the future are briefly described in Section 7.

With an optimal set of sentences (or phrases) as input, the **text reformulation** stage is concerned with reconstituting the extracted sentences into a coherent summary. Thus, a chronology will have to be established (where possible, this will be the same ordering as used in the source documents); anaphors will have to be resolved to ensure that any pronouns used have a referent in the summary; and in the case of multi-document summarisation, contradiction resolution may be required, if two or more source documents contain conflicting "facts".

3 Multi–Document Summarisation

The multi–document summarisation process has a number of extra tasks to perform. For example, **sentence ordering** will generally be required to impose a coherent stucture on a summary extracted from multiple sources. While it is trivial to order sentences from a single document (just use original ordering), this is not the case with multiple sources. In some cases, such as topic tracking, the summaries need to reflect the changes to the story with respect to time. Hence, ordering can be based on the order in which the source documents were published.

However, in many cases there are multiple texts written about a single event. In these cases it can be very difficult to get a reliable ordering on sentences because there is no order on the source documents. Instead, an ordering is derived from any textual clues that may be available, such as references to times and dates, days of the week, or phrases such as "yesterday", "last week", "in the next year". From these clues, partial orderings can be placed on the sentences, from which at least some coherence is guaranteed.

As with sentence ordering, **contradiction resolution** is a very important, but also potentially very difficult part of Multi–Document Summarisation. Articles written by different groups, from different viewpoints, or at different times, may differ significantly in the information presented. Not only is there the problem of subjective opinions to deal with, but also simpler problems such as facts which genuinely do change over time (e.g. a series of articles about a natural disaster such as an earthquake will probably have an ever-rising "death toll"). Some of these conflicts may be ameliorated by decent chronological ordering (indeed, conversely, a trend such as a rising "death toll" may be used by an ordering process).

Redundancy removal is an integral part of a Multi–Document Summarisation system and is discussed in detail in Section 4.

4 Redundancy Removal Techniques

For the purposes of this paper, we focus exclusively on the redundancy removal module of our summarisation architecture. The most important part of a redundancy removal process is

its *similarity measure*. In this section, we describe the similarity measures under investigation in this experiment.

There are many different ways of computing the similarity between two texts. *SimFinder* [5] is a clustering tool that finds the similarity between text units using a statistical measure which incorporates various linguistic features. Barzilay [6] describes a very interesting rule–based approach to paraphrase identification.

For our purposes, we are focusing on two vector–based methods and a simple WordNet–based similarity measure.

WordNet Distance In this WordNet–based measure, we define the similarity between two sentences as the sum of the maxmimal similarity scores between component words in the WordNet, using the Leacock-Chodorow measure [10, 11].

To measure the similarity between a pair of sentences, we examine every word in both sentences, and take the sum of the highest word–pair scores as the similarity score for the sentence.

Cosine Similarity A simple *zero–knowledge* method for measuring sentence similarities is to use a vector–based method such as cosine similarity [3]. We use a vector-space model [12] as the primary data structure in the Cosine Similarity and Latent Semantic Indexing measures. Sentences are stopped and stemmed using the Porter algorithm [13], and a count of all the words used in the sentences of the corpus is calculated. This count provides us with the information to construct the *termspace*, an n-dimensional vector space, where n is the number of unique terms found in the corpus.

With a vector representation of all of the sentences of the corpus, we can take a simple measure of the similarity of any pair of sentences by looking at the size of the angle between their vectors; the smaller the angle, the greater the similarity.

Latent Semantic Indexing Latent Semantic Indexing [14, 15] takes as input a term–document matrix constructed in exactly the same way as for Cosine Similarity. Before applying a similarity measure between the vectors, an *information spreading* technique known as Singular Value Decomposition is applied to the matrix.

This technique scrutinises the term–document matrix for significant levels of word co–occurence and modifies magnitudes along appropriate dimensions (i.e. scores for particular words) accordingly. Thus, a sentence such as “The Iraqi leader was deposed” may have its vector representation modified with increased magnitude along dimensions corresponding to the terms “Saddam Hussein”, “Baghdad”, and “George W. Bush”, for example.

In terms of matrices and vectors, SVD can be crudely expressed as finding the dimensions which lie close together (co–occur) and compressing them onto one composite dimension. This means that vectors which may be closely related (in terms of information content or topic) but contain no common words (and therefore would not have a high similarity rating under Cosine Similarity measure) would be recognised as similar under LSI analysis.

5 Evaluation Methodology

The corpus we used for evaluation was derived from the 2003 DUC corpus. The Document Understanding Conference (DUC) [1] provides a testbed of newswire documents and ma-

1a - Danforth named to lead independent query of FBI siege at Waco.

1b - Former Senator Danforth named to head investigation of Waco siege.

Figure 1: Mutually redundant sentences

chine translated documents that groups can use to evaluate their systems and compare their performance to other state–of–the–art approaces.

Within the programme, there are a number of tasks which can be undertaken, such as the generation of very short single–document summaries, multi–document summarisation and question–answering.

For this paper, we developed a corpus based on the very short, single–document *human–generated* model summaries, provided in the DUC corpus for evaluation purposes. We examined the models for each story and selected those pairs of models which were deemed to contain the same information, i.e. are paraphrases of each other. Given the purpose of these models in the corpus, the number of sentences which were judged to fit this criterion is surprisingly small. We extracted only 404 sentences from the corpus (ie 202 pairs of similar sentences) from a pool of approximately 1300. An example of the sentences judged to be paraphrases is given in Figure 1.

5.1 Evaluation Methods

For this application, the most relevant evaluation metrics are those used in the TREC/TDT tasks. These are the traditional IR metrics of Precision and Recall.

Precision is a measure of the proportion of relevant sentences that were retrieved. **Recall** is a measure of the proportion of retrieved sentences that are relevant. Thus, a high precision score means that most of the retrieved sentences are relevant, and high recall implies that most of the relevant sentences in the corpus were retrieved.

6 Results and Conclusions

Figure 2 shows how precision varies with respect to recall for the redundancy techniques described in Section 4. We tested the LSI method at a number of different resolutions (rank reductions), but as all results for that system were highly similar, we plotted only one on the graph.

The poor Precision/Recall performance of WordNet can be accounted for by its very high error rate (not presented here). Looking at the data, we found this high error rate can probably be traced to the fact that WordNet maps a word to a potentially large number of different senses. In our method, we make the assumption that the highest scoring sense is most likely the correct one. Unfortunately, that assumption can lead to mis–matches, and thus we find the method makes many false positives, i.e. it identifies non–matching sentences as being similar.

We can see that there is no difference in performance between Cosine Similarity and Latent Semantic Indexing. Because of its information–spreading capability, we expected the LSI method to perform more strongly than the others. However this did not turn out to be the case. An analysis of the results suggests that the size of the corpus is the cause of the poor performance of LSI. There were not enough examples in the corpus to provide the method with the data necessary to find significant co-occurences. Thus, the LSI module could not

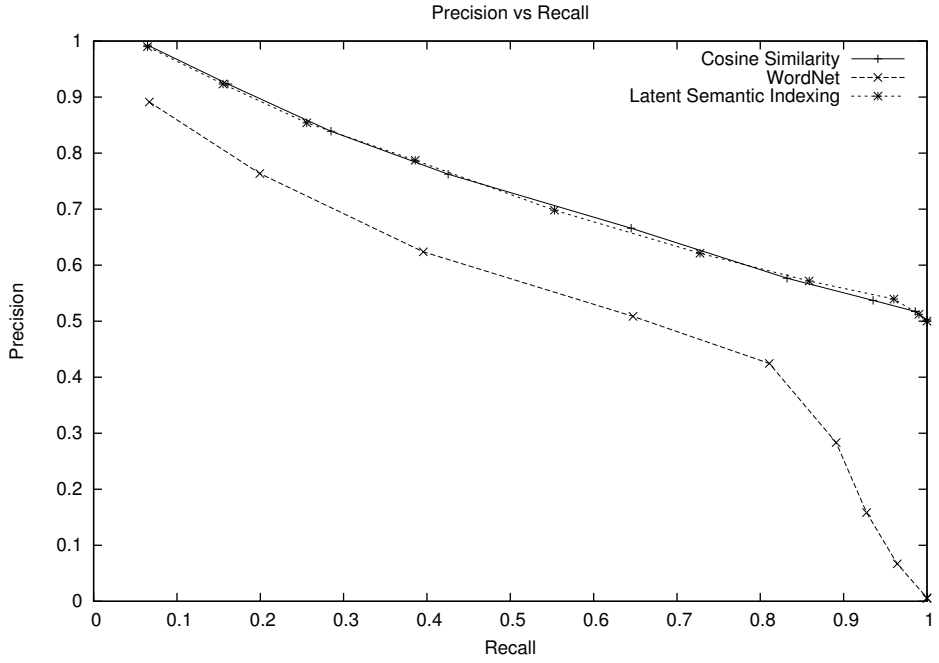

Figure 2: Graph of Precision against Recall for the 3 metrics under examination.

make any significant improvements to its index, and therefore did not perform any better than the baseline Cosine Similarity measure.

Further experiments are being carried out on an expanded corpus to improve the co–occurrence data presented to the LSI module. Unfortunately, these experiments were not completed in time for inclusion in this paper.

7 Future Work

Our future research in this area will take into consideration a wide range of different methods. Schiffman at al. [16] describe a system which uses a much wider set of features than our system for determining possible candidate sentences for the summary. These features cover a wide range of textual properties, from semantic aspects to features like publication date, or the position of a sentence in the text. One noteworthy feature is *verb specificity*, where particular verbs carry more information simply because they usually only co-occur with a small set of significant nouns (e.g. “arrest” with “police”). These verbs are much more useful in a summary than more general verbs such as “do” or “go”.

Maximal Marginal Relevance [17] also provides some interesting study. The paper describes a system which ranks retrievals made by an IR system, in response to a query. They attempt to eliminate redundancy in the set of retrieved texts by giving a low score to any texts which are highly similar to previously–seen items. Thus, it is ensured that there is minimal overlap between the highest–ranked retrievals.

Allan et al. [18] describe a system which attempts to make temporal summaries of news streams, i.e. each summary captures only the novel aspects of the story as it evolves over time. Thus, this system requires a means of identifying what is redundant and what is novel with respect to what has gone before. This presents a further challenge for redundancy removal techniques since there is no scope for reconstructing summaries at a later stage when

subsequent news stories have been analysed, due to the temporal nature of the task.

References

[1] National Institute of Standards and Technology, *DUC: Document Understanding Conference*. At *http://www-nlpir.nist.gov/projects/duc*.

[2] H.P. Luhn, *The Automatic Creation of Literature Abstracts*, in "Advances in Automatic Text Summarization", eds. Inderjeet Mani and Mark T. Maybury. Originally in *IBM Journal of Research and Development*, April 1958.

[3] C. J. van Rijsbergen, *Information Retrieval*. At *http://www.dcs.gla.ac.uk/Keith/Preface.html*.

[4] Daniel Marcu, *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts*, PhD Thesis, University of Toronto, 1997.

[5] Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, Regina Barzilay, Min-Yen Kan, and Kathleen R. McKeown, *SimFinder:A Flexible Clustering Tool for Summarization*, NAACL Workshop on Auromatic Summarization, Association for Computational Linguistics, 2001.

[6] Regina Barzilay, *Information Fusion for MultiDocument Summarization: Paraphrasing and Generation*, PhD Thesis, Columbia University, 2003.

[7] William P. Doran, Nicola Stokes, John Dunnion, Joe Carthy, *Assessing the Impact of Lexical Chain Scoring Methods and Sentence Extraction Schemes on Summarization*, Proceedings of 5th Int. Conf on Intelligent Text Processing and Computational Linguistics, 2004.

[8] National Institute of Standards and Technology, *TREC:Text REtrieval Conference*. At *http://trec.nist.gov*.

[9] Hamish Cunningham, *GATE: A General Architecture for Text Engineering*. At *http://gate.ac.uk*.

[10] George A. Miller et al., *WordNet: Lexical Database for the English language*, Cogniive Science Laboratory, Princeton University. At *http://www.cogsci.princeton.edu/∼wn*.

[11] Alexander Budanitsky, *Lexical semantic relatedness and its application in natural language processing*, Technical Report, University of Toronto, 1999.

[12] R. Baeza-Yates, B. Ribeiro-Neto, *Modern Information Retrieval*, ACM Press, 1999.

[13] Martin Porter, *An Algorithm for Suffix Stripping*, in Progam, vol. 14, no. 3, July 1980. At *http://www.tartarus.org/∼martin/PorterStemmer/def.txt*.

[14] S. Deerwester, S. T. Dumais, G. W. Furna, T. K. Landauer and R. Harshman, *Indexing by Latent Semantic Analysis*, Journal of the American Society for Information Science, 1990.

[15] T. K. Landauer, P. W. Foltz and D. Latham, *Introduction to Latent Semantic Analysis*, Discourse Processes, 1998. At *http://www.knowledge-technologies.com/publications.html*.

[16] Barry Schiffman, Ani Nenkova, Kathleen R. McKeown, *Experiments in Multi–Document Summarization*, Proc. Conf. Human Language Technology, 2002.

[17] Goldstein, Mittal, Carbonell, Kantrowitz, *Maximal Marginal Relevance*, Proc. ANLP, NAACL Workshop on Automatic Summarization, 2000.

[18] James Allan, Rahul Gupta and Vikas Khandewal, *Temporal Summaries of News Topics*, Proc. SIGIR 2001.