An Approach to Summarisation Based on Lexical Bonds

Murat Karamuftuoglu

Visiting Researcher
Microsoft Research Cambridge
hmk@soi.city.ac.uk

Introduction

Our objective in DUC 2002 was to investigate the use of extraction oriented statistical and pattern matching methods, in particular lexical links and bonds, in creating single documents summaries. In what follows below, we describe the main methods used in our participation in DUC 2002 and some preliminary experiments conducted to gauge the effectiveness of it.

The Strategy

Our strategy was based on the 'Extract – Reduce – Organize' paradigm (Figure 1). In other words, the strategy was first to extract sentences that are most representative of the original text, then apply text compaction methods to reduce the number of words needed to express a given idea or piece of information. Anaphoric references, more specifically, pronouns were also to be replaced with their referents at this stage. The remaining pieces of text were then to be put together to produce a coherent summary.

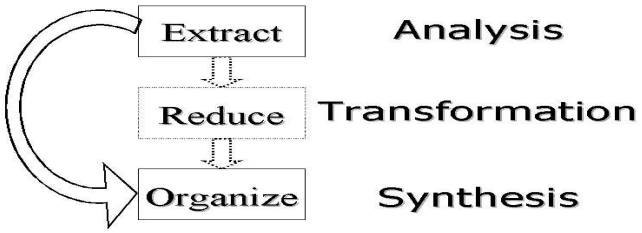

Figure 1: Summarization Strategy

The analysis and synthesis stages are described in more detail below. The text transformation part of the system (both text compaction and anaphora resolution), however, was not completed in time for the DUC 2002 submission. Therefore, in our submission extracted sentences from the original document were used in the summary as they appeared in the original without any modification.

Analysis

The system pre-processed the original documents and recorded surface linguistic information about each of them (Figure 2).

Figure 2: Analysis

Pre-process. The original documents were broken into sentences using the DUC software. Stop words were removed based on a short list of functional words (e.g. auxiliary verbs, articles). Pronouns were retained. The remaining words were stemmed using the Porter stemmer [1].

Record surface linguistic features and information content. Table 1 presents the information recorded for each sentence in each document. Most of these features are self-explanatory. Feature 3 is Boolean, which is used to distinguish the first and last 5 sentences in the document from the others. These sentences are thought to be more likely to be included in the summaries than the others. Lexical link and bond counts are done as described below. Similarly the calculation of BM25 scores are described below.

Feature no.	Type	Description
1	integer	Sentence position in the document from 0 for the first sentence to *n-1* for the last, where *n* is equal to the total number of sentences in the document.
2	integer	Total no. of sentences in the document
3	Boolean	First and last 5 sentences get 1 others 0.
4	integer	No. of words in the sentence.
5	integer	No. of lexical bonds the sentence has with the sentences before it in the document. Bond threshold = 2 lexical links (words/stems).
6	integer	No. of lexical bonds the sentence has with the sentences after it in the document. Bond threshold = 2 lexical links (words/stems).
7	integer	Total no. of lexical bonds the sentence has with the sentences in the document. This is equal to the sum of features no. 5 and 6. Bond threshold = 2 lexical links (words/stems).
8	integer	No. of lexical links the sentence has with the sentences before it in the document (i.e. total number of words/stem that the sentence shares with the sentences before it).
9	integer	No. of lexical links the sentence has with the sentences after it in the document (i.e. total number of words/stems that the sentence shares with the sentences after it).
10	integer	Total no. of lexical links the sentence has with the sentences in the document (i.e. total number of words/stems that the sentence shares with the other sentences). This is equal to the sum of features no. 8 and 9.
11	integer	Total no. of lexical bonds the sentence has with the sentences in the document. Bond threshold = 1 lexical link (word/stems).
12	real	BM25 weight (b=1)

Table 1: The Feature Set used in the Sentence Selection Process.

Calculate lexical links & bonds. A lexical link between two sentences was defined as a word (stem) that occurs in both sentences. Two or more lexical links between a pair of sentences constituted a lexical bond between them [2].

Calculate sentence score (BM25). To compute a score, which is indicative of the information content of the sentence or its importance in the document, we created a database of single sentences using the Okapi retrieval system [3]. Each record or document in this database was a single sentence. We ran the original documents as queries against this database. A matching score based on the BM25 function [4] was computed for each of the sentences in each document. The scores are normalised by sentence length (BM25 parameter b was set to 1). This score was taken as the indicator of the importance or informativeness of the sentence in the corresponding document and used in the sentence selection process (see below).

Sentence Selection

We used a machine learning system based on Support Vector Machines (SVMs), namely SVMLight [5], to select sentences from the source document for inclusion in the summary. Each sentence is represented by the feature set given above. We trained the system with half of the set of manually selected extracts from last year's test documents, provided for training purposes in DUC 2002 by John Conroy. The other half of the set of extracts was used to tune the parameters of the learner. We used a linear decision function (t=0) with the parameters c=1 and j=10. Our results gave average precision of 38.83% and of recall 42.11% with this set of extracts.

Figure 3: Sentence selection

Sentence Selection by Lexical Bonds

We also experimented with generating summaries by following lexical bonds from a given source sentence. The starting or source sentence in our experiments was the first sentence in the main body of the document that has at least one forward lexical bond (i.e. has at least two tokens in common with a sentence that comes after it in the text). Figure 4 illustrates this process. In the example given, each sentence is identifies by the set ID, document ID and a sequential sentence number, shown in bold. The next two digits in the figure indicate the number of backward and forward bonds the sentence has respectively. The numbers that follow the colon in the figure are the sentence numbers that form a bond with the sentence in question. In the figure backward bonds are separated from the forward ones by a backslash.

<d109h><FBIS4-26604><0> **0,7:** \ 1, 3, 4, 5, 7, 10, 11,
<d109h><FBIS4-26604><1> **1,5:** 0, \ 4, 5, 7, 10, 11,
<d109h><FBIS4-26604><3> **1,5:** 0, \ 4, 6, 7, 10, 11,
<d109h><FBIS4-26604><4> **3,4:** 0, 1, 3, \ 5, 6, 10, 11,
<d109h><FBIS4-26604><5> **3,2:** 0, 1, 4, \ 10, 11,
<d109h><FBIS4-26604><10> **7,1:** 0, 1, 3, 4, 5, 6, 7, \ 11,

Branch 1: 0,1,4,5,10,11 → bm25 score (document as the query)
Branch 2: 0,3,4,5,10,11 → bm25 score (document as the query)
……

Figure 4: Branches by Lexical Bonds

A summary was formed by following lexical bonds one by one from the source sentence to the one lexically bonded with it, and from that sentence to the other which has a bond with it, and so on. In Figure 4 a few of the possible branches that could be geneated from the given bond structure are illustrated. Obviously, there are far too many possible combinations available in this way and several megabyte of branches were generated for even relatively short documents in our experiments. We therefore seeked to reduce the number of combinations (branches) by imposing some constraints on the sentences to be included in the branches. We only followed from the source sentence those sentences that are in the upper half of the document (i.e. those with sentence numbers less or equal to the half of the biggest sentence number in the document), and also only sentences selected by SVMLight are included in the branch generation process. Although these two constraints significantly

dropped the amount of branches generated we still had too many combinations (some 80,000 branches or about 10 Mb for 40 documents taken from the last years test collection).

For a small subset (40) of the documents available for training we generated clusters of sentences (branches) in the way described above. We then constructed a database of branches using the Okapi system. Each branch was given a matching score based on the BM25 function using the corresponding document as the query as discussed earlier. The highest scoring branch for each documment was selected to form the summary for that document. Our experiments gave an average macro precision of about 30% and recall of 19% using the small sample of 40 documents. However, the branches with much higher concentration of good sentences – even after imposing the constraints mentioned earlier – were among those generated by this method and if it were possible to identify them the average precision would be around 71% and recall 45%. Obviously the BM25 scores as computed in the experiments were not able to identify the best branches. We believe that more experimentation in this direction could reveal more effective ways of reducing the number of branches generated and better ways of selecting good ones.

Synthesis

In our submission, we selected sentences by SVMLight, as described earlier. A summary is formed by taking the sentences selected by the learning system starting with the one that appears first in the document. The next selected sentence was included in the summary *iff* it has a lexical bond with at least one of the previous sentences included in the summary thus far. An exception to the rule was that if the SVM-selected sentence had no backward bonds, it was still included in the summary. The argument for this exception was that such sentences might be new topic opening sentences (i.e. introduce a new topic or information) and it would be useful to have them in the summary. This process was repeated until the 100-word limit was reached.

Summary and Discussion

The main advantages of the method described above are that it is based on relatively simple statistical and pattern-matching operations and the sentences that are lexically bonded should yield reasonably coherent summaries. The main disadvantage is that the extracts formed in this way are usually very verbose and therefore not very effective in representing a given amount of information in a smaller amount of space. Various methods of text compaction (e.g. disembedding relative clauses) should help to reduce the number of words to express a given information.

Another point of concern, especially for short summaries of 100 words or so, is that in the method we described the summaries generated tend to be dominated by sentences from the earlier parts of the document. This is especially a problem when the sentences from the original text are used without any modification. In our submissions we found that 58% of the summaries were formed by sentences that sequentially follow each other in the original text. Only 42% of the summaries had a gap between two subsequent sentences (i.e. contained sentences that are not consecutive in the original).

Finally, the alternative method of generating clusters of sentences by following lexical bonds that spread out from a given source sentence described earlier seems to hold a promise in increasing the precision of the generated summaries. The main problems in this method were the huge number of possible combinations (branches) generated in this way and identifying

the best branches available. Further research along these lines could help produce better extract-based summaries.

References

1. The Porter Stemming Algorithm. http://www.tartarus.org/~martin/PorterStemmer/
2. Hoey M. *Patterns of lexis in text.* Oxford University Press, 1991.
3. Okapi Home Page. http://web.soi.city.ac.uk/research/cisr/okapi/okapi.html
4. Sparck Jones K., Walker S., Robertson S. *A Probabilistic Model of Information Retrieval: Development and Status.* University of Cambridge Computer Laboratory Technical Report N 446, 1998.
5. SVMLight. http://svmlight.joachims.org/