Incremental Retrieval of documents relevant to a topic

Caroline Lyon, Bob Dickerson, James Malcolm
University of Hertfordshire, UK
c.m.lyon@herts.ac.uk

Introduction

As new participants to TREC, on the Filtering Track, we have started by first investigating two methods of producing document profiles. We begin by looking for "obvious" profiles that detect closely related documents. This year we have started by looking for:

- lexically similar cases
- semantically similar cases based on a simple combination of keywords.

Characteristics of the Reuter's data

Before addressing specific tasks we investigated the Reuter's data. It was expected in this domain that there would be some similar text in different documents: the extent is quite significant. We used the Ferret software, designed to ferret out similar passages of text in large document collections, which we have recently developed [2].

An experiment was carried out to compare each document with about 1000 others, taken in date order. We went through the test corpus (723141 documents) and for every set of 1000 documents compared each with each (that is 499500 comparisons for each set). Of course if file A is similar to file B and to file C, then it is quite likely that File B is similar to file C.

We found 48,918 with identical text. Some of the files were very short, for instance regular industrial reports might have no more than 10 content words in the text. Omitting files with 10 or less content words, 6,616 had identical text.

The analysis also showed that in a further large number of file pairs texts were "very close" – this and other terms will be explained below. 287,391 pairs fell into this category. Without those files containing 10 or less content words in their texts, 24,017 were very close.

There are 718, 443 pairs with "significant matching passages". Of those with more than 10 content words in the text 228,130 fall into this category.

Method of determining similarity

The method used is as follows. First each document is pre-processed so that only the id number, the headline, and the text are kept, while tags are omitted. Stop words are filtered out. There are 440 stop words, and the list includes entries which, though not function words, have little semantic content.

Then each document is converted into a set of word triples, composed of every sequential triple. Thus, the sentence:

Given a topic description and some example relevant documents build a filtering profile.

would be converted into the set:

given a topic a topic description topic description and etc.

or, after taking out stop words:

given topic description topic description example description example relevant etc.

Then each pair of documents is compared for matching word triples. This raw score is converted into the metric "resemblance", based on set-theoretic principles. Informally, resemblance is the number of matches between two sets, scaled by joint set size. It is also known as the Jaccard coefficient. Let S(A) and S(B) be the set of trigrams from documents A and B respectively. Let R(A,B) be the resemblance between A and B

$$R = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|}$$

For the preliminary investigations into the Reuter's data, documents are identical if, after pre-processing, R =1.0. The category "very close" takes $1.0 > R \geq 0.8$, while "significant matching passages" takes $0.8 > R \geq 0.4$. These are arbitrary boundaries.

As an indication of the scale of similarity, it is worth considering measures used in another field. The Ferret was originally developed for detecting plagiarism in students' work. At a level of $R > 0.04$ (a degree of magnitude smaller than that used here) matching passages were typically found, possibly quite short.

Time taken to process each set of 1000 files was about 1 minute, about 11 hours for the full test set, on a Pentium III processor, with 700MHz, 512 MB RAM. However there is considerable scope for increasing the efficiency of this implementation.

Theoretical background

The dominant approach in statistical pattern analysis is based on the well known method of abstracting significant features and lining them up in a feature vector for further processing. However, there are relationships between the number of elements of the feature vector, the amount of training data available and the level of generalization achieved. In text processing a very large number of words have to be processed, even after filtering through a stop word list. The amount of training data will typically not be enough to ensure a satisfactory level of probably approximately correct outcomes. For further details see [1, 3]. Therefore, a set theoretic approach may be appropriate in word based text processing, as described in [2].

Routing filtering with lexical profiles

The method described above was then applied to give a preliminary analysis of topics in the filtering task. For this we just took the three sample documents given for the adaptive filtering task, and did not refer to the topic description. The three sample documents are stripped of xml tags, edited by filtering through the stop word list and concatenated. This text is then compared to all the documents in the test data (similarly detagged and filtered through the stop word list). For Topic 102 a pairing producing 16 matches, resemblance 0.05, is displayed, Figure 1. The number of matching word triples shown in the display is much greater than that produced by the match detection software, since for display we go back to the original documents which include stop words and xml tags.

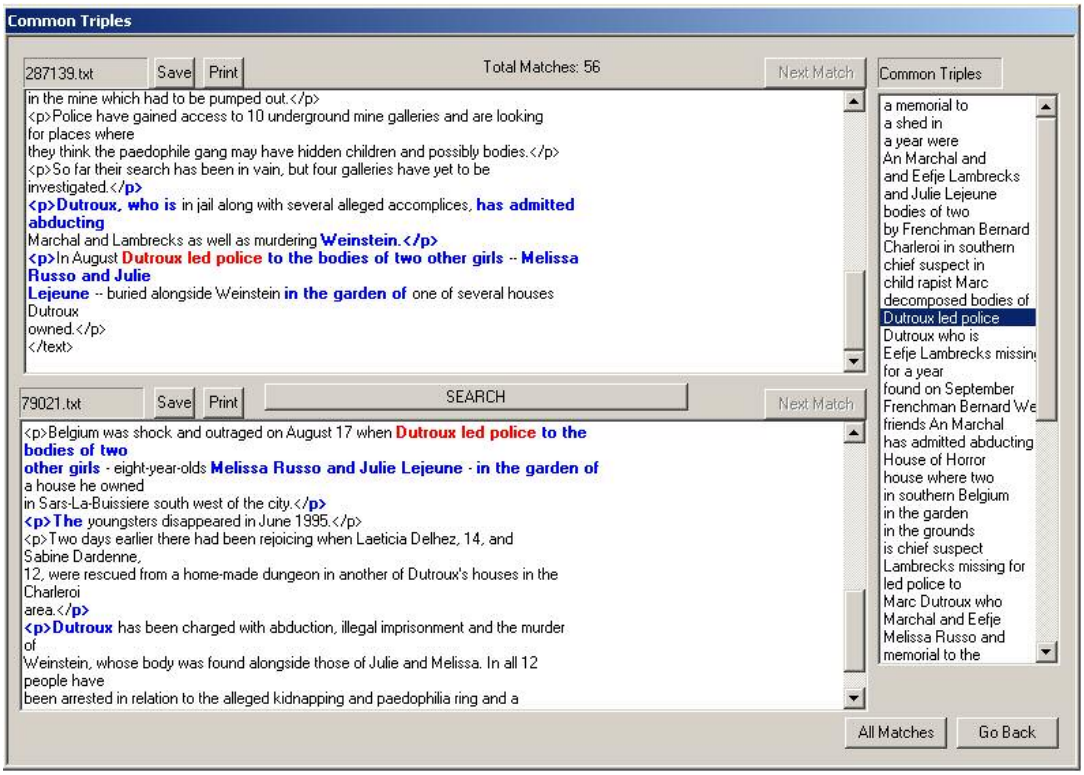

Figure 1: From Topic 102, display of sample text 79021 and relevant document 287139

In Figure 1 the lower of the two files, id 79021, is one of the 3 example documents for Topic 102. The upper file, id 287139, has short passages of matching text. It seems that the original story was picked up again some time later.

This illustration shows how short passages of matching text can be detected. Lexically similar text is often semantically similar too. However, this is not always the case, as when the processor picks up commonly occurring comments such as "Reuters has not verified these reports and cannot vouch for their accuracy".

The type of lexical similarity described above indicates semantic similarity. However, the opposite is not true. If two people write on the same topic independently the resulting articles will not be lexically similar in this way, as previous experiments have shown. When texts are lexically similar it indicates that there has been some element of cutting and pasting.

Routing filtering with simple keyword profiles

The concept behind this method is to have several sets of keywords, and for a document to be considered relevant it must have at least one member in each set. The keywords have been selected manually at this point, from the topic description and three sample documents for the adaptive filtering task. The rest of the training data was used for primary evaluation of this approach. Topics R101 and R125 were entered on this track. For initial work there were 3 sets of keywords. It was essential to have a member in sets key1 and key2. Key3 was a set of supporting keywords whose frequency of occurrence determined the ranking. As an example, the keywords for Topic R101 on industrial espionage were as follows:

key1	key2
espionage spy spying	business commercial economic industrial technical

Figure 2 : Essential keywords

Using this method cuts down on possible combinatorial explosion of combinations of terms "industrial espionage", "commercial espionage", "industrial spying" etc. On inspection later, it seemed that key1 might have included "secrets" and key2 "company". This would have caught some documents that slipped through the net, but might have produced false positives too.

key3	
charges	police
confidential	prosecution
court	prosecutor
courts	prosecutors
covert	secret
intelligence	secrets
investigation	surveillance

Figure 3: Non-essential keywords used for ranking

Results

Using this method on topic R101 produced a score of 0.428 compared to median 0.469 and maximum0.902. On R125 it produced a result of 0.062, compared to a median of 0.327 and maximum of 0.565.

In both case the number of relevant documents was well below the specified number. For R101 477 were found, for R125 260 were found. However, limited random sampling indicated that no false positives were found.

Discrepancies in the data

In some cases the topic description and the training documents were not consistent. For example, Topic R110 was entitled “Terrorism Middle East tourism”, and the narrative said relevant documents should correlate terrorism with tourism. However, “terrorism” and associated terms were not mentioned in the 3 training documents for adaptive filtering (42439, 82926, 85147). Topic R125 was entitled “Scottish Independence” but there was no mention of Scotland in any form in some documents judged relevant (27974, 48375, 68664). In Topic 134 the narrative of the topic description said that documents were relevant only if statistics were included. There were no statistics in one of the three training documents (73372).

Conclusion

The first method employed detected little of that lexical similarity between training and testing documents, which is indicative of re-using text. However, our investigation of general characteristics of the data showed that there is much re-use of text on close dates.

Taking a sideways glance at the Novelty Track, this method could be useful to sort out similar versions of a story from ones with new information. Whether the new information is strictly relevant would be another matter. For instance, reports on ABA banking policy (100017,100398) had similarities (resemblance 0.55). The second had additional information, on the speakers’ clothes, which might not be considered relevant.

The second method employed, using combinations of keywords, is a useful way of a detecting a core of relevant documents. This could possibly be automated using thesauri and/or Wordnet.

If the Filtering track is reinstated we plan to move on to the more interesting hard-to-detect cases, and to integrate different profiles as in co-training.

References

1. A K Jain, R P W Duin and J Mao. Statistical Pattern Recognition: A review. *IEEE Trans. on Pattern Analysis and Machine Intelligence, 22, 1. 2000.*

2. Caroline Lyon, James Malcolm, and Bob Dickerson, Detecting short passages of similar text in large document collections, *Proc. of Conference on Empirical Methods in Natural Language Processing,* 2001.

3. C Lyon and R Frank. Using single layer networks for discrete sequential data: an example from Natural Language Processing. *Neural Computing Applications 5 (4) 1997*