Feature Reduction for Information Retrieval

Stefan M Rüger
Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate, London SW7 2BZ

1 Introduction

Our experiments for the ad hoc task were centred around the question how to create a document surrogate that still contains enough information to be used for a high-quality, efficient retrieval.

In the first step we drop all the function words and all the auxiliary words that although having a proper meaning merely help to communicate about the topic without being relevant to the topic. We apply part-of-speech analysis in order to retain the nouns and adjectives of a document. Standard term and document frequency analysis is used to compute a weight factor for each of the remaining words.

In a second step, we plan to set the relevant words into a relation that conveys a part of the meaning. Like in vector space models, both topic and document would be represented in this keyword-relation form and a suitable metric would quantify the relevance of a document to a topic.

At this stage of our research, no relations are stored in document surrogates. The automatic processed topic descriptions, however, include some very crude relation analysis that, eg, transfers "relevant documents describe cases of drink-driving outside France" to "drink driving outside France" and hence, knowing about the connotation of " ... outside ...," a negative weight factor for the occurrence of drink-driving and France. It is planned for future work to analyse relations more and more with statistical models and with trained probabilistic models and less with linguistic analysis.

For now, the purpose of our experiments is assessing the performance of the above very simple model of pure feature reduction without relations, without training/learning weights without sophisticated recall procedures, without inverted document files and without a proper document retrieval system. It might be interesting to see which effect feature reduction algorithms have in other, sophisticatedly tuned systems.

2 Preprocessing of the Documents

2.1 Data Flow

The basic assumption is that a document collection consists of one or more files. Each file contains one or more `SGML/XML`-tagged document bodies that are each preceded by the line

```
<DOCNO> document-name </DOCNO>
```

The preprocessing process is shown in Figure 1. In the moment, several steps are involved. Most notably, a part-of-speech tagger `pos-tagger` (Brill 1994) is used to find the role of each

word. We believe that nouns and adjectives are most vital to the contents of a document. Hence, we only consider words (including proper nouns) that are used in this way. We eliminate stop words, fold all characters to lower case and use Porter's stemming algorithm to obtain word stems (thus, eg, identifying singular words with their plural form). A simple analysis associates a document with a list of relevant word stems (these are the *terms* in our context), and their term and document frequency.

In a heuristic attempt to increase the weight of titles, headlines, etc, we count each word fourfold that is enclosed in a matching XML command pair at the beginning and end of the same line such as `<TITLE> Cuba Crisis Revisited </TITLE>`.

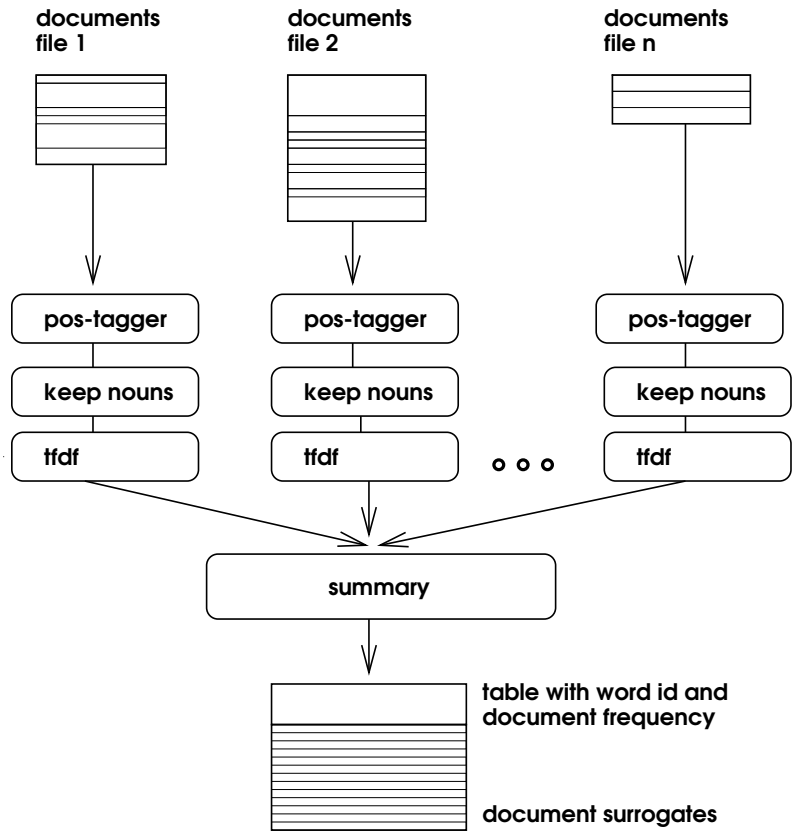

Figure 1: Preprocessing: data flow

The vocabulary (set of different words or terms) of a growing document collection does not seem to saturate even at a high number of documents. Figure 2 shows the number of different words versus the number of words in a growing document collection (up to 210,000 articles, 4 years of Financial Times).

As the vocabulary increases, so does the set of adjectives and nouns that are left after preprocessing. We verified that the validity of Zipf's law (Zipf 1949) also extends to the subset of nouns and adjectives. Hence, most of these words are only used in one document. We compute a histogram of the document frequencies (this histogram maps the document frequency to the number of nouns that have this document frequency). The basic idea is that *potentially meaningful* words occur with medium document frequency. Hence, we disregard words that occur in only one document or that occur in more than half of the documents.

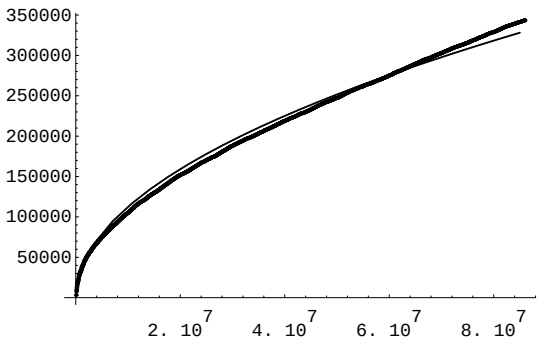

Figure 2: Number of different words vs number of words in a growing document collection (fat line, 4 years Financial Times) and the square root in comparison (thin line).

The output of the whole preprocessing is *one* summary file that contains a table of the relevant words together with an id number and the document frequency, respectively. This file also contains a document surrogate of each document, ie, a collection of the relevant words together with the term frequency of this word in the collection.

The retrieval process is based on this summary file. Owing to the lack of a retrieval system at the time of the experiments no inverted document list was available. So, in order to retrieve documents, the whole summary file had to be scanned, resulting in an inferior query time of 10 seconds per query for the TREC-7 task.

2.2 Complexity and Resources for the Document Preprocessing

Let m be the number of different words in the document collection and n be the number of words in the document collection. The time for preprocessing is predominantly linear in n and the space for internal arrays is predominantly linear in m. Theoretically, the time is of order $O(mn)$, but a clever use of hash-tables or other methods can disguise the m dependency. The preprocessing throughput is around 12 Megabyte/hour of documents on a typical Sparc workstation. It should be borne in mind that the prototype was not tuned for speed and that the algorithms used are highly parallelisable. (Remark: in the meantime, a tuned all-in-one version of the same software achieves a throughput of well over 300 Megabytes/hour.)

3 Processing of the Topics

The topics are processed in a similar way as the documents are preprocessed. The vocabulary of the topics is, however, not limited by document (or topic) frequencies, as is the case with the document collection. The internal structure of the topics are exploited: The title and description part are processed as discussed before, and a corresponding list of relevant words together with their term frequency (how often this word is used as an adjective or noun in this topic) is associated to this topic. This list is called `r-list`.

For the run `ic98san3`, this list of words is matched to the list of relevant words of each document and a ranking number is computed for each document using the set of common words as explained below. The 1000 best-ranked documents are the result of this run, which does not make use of the narrative field of the topics.

The run `ic98san4` re-ranks the 1500 best-ranked documents of the `ic98san3` run. Here two

new lists, the `p-list` and `m-list`, are created per topic: the narrative field is examined and all clauses are analysed w.r.t. the relevance to the topic. This is done using a set of phrases such as "* is relevant, but * not relevant". Adjectives and nouns that appear in clauses relevant to the topic are added to the `p-list`, and those that appear in clauses that are explicitly not relevant to the topic are added to the `m-list`. In the end, the elements of the `r-list` from the `ic98san3` run are added to the `p-list`. Both lists are matched to each of the previously 1500 best-ranked documents and two corresponding ranking numbers are computed per document. Their difference is used to re-rank the 1500 "best" (according to the `ic98san3` run) documents per topic. The corresponding 1000 best-ranked documents are the result of this `ic98san4` run.

4 Relevance Assessment

This section describes how relevance numbers are computed given the `r-`, `p-` and m-lists of a topic. Let n be the number of documents, i be one of the documents and j be a term (in our case lowercase word stems of words that are used as adjectives or nouns in a document or a topic). Let V_i be the vocabulary of document i, let $T^{r,p,m}$ and $t_j^{r,p,m}$ be the vocabulary and the j-term frequency of the corresponding topic lists. Let d_j be the document frequency of term j and t_{ij} be the term frequency of term j in document i. Then,

$$w_i^r := \left(\frac{|V_i \cap T^r|}{|T^r|}\right)^4 \cdot \frac{1}{(1+|V_i|)^2} \cdot \sum_{j \in V_i \cap T^r} t_{ij} t_j^r \log(n/d_j)$$

is the ranking number that is used for run `ic98san3`.

For `ic98san4` the 1500 best-ranked documents of the `ic98san3` run are re-ranked with the ranking number $w_i^p - w_i^m$, where

$$w_i^{p,m} := \left(\frac{|V_i \cap T^{p,m}|}{|T^{p,m}|}\right)^3 \cdot \frac{1}{(1+|V_i|)^2} \cdot \sum_{j \in V_i \cap T^{p,m}} t_{ij} t_j^{p,m} \log(n/d_j).$$

5 TREC Evaluation and Conclusions

TREC assigned an average precision over all relevant documents of 0.1259 to the run `ic98san3` and 0.1333 for `ic98san4`, respectively. These numbers are not very impressive in comparison to other system's performance, and reflect the rather basic structure of the whole retrieval process.

Our experiments were concerned with the *feature reduction* component of a whole system. It is interesting to note that evaluating the description part of the topics increased the average precision slightly, as would be expected from a sensible feature extraction algorithm.

The next natural steps of a better retrieval performance would be to incorporate a standard inverted-document list retrieval to obtain a sensible query speed, optimise the ranking procedure, implement relevance feedback and reassess the system.

References

Brill, E. (1994). Some advances in rule-based part of speech tagging. In *AAAI*.

Zipf, G. K. (1949). *Human Bahavior and the Principle of Least Effort*. Addison-Wesley.

Acknowledgements: This work was supported by the Fujitsu European Centre for Information Technology (FECIT).