Example Based Text Matching Methodology for Routing Tasks

Ari Visa, Jarmo Toivonen, Tomi Vesanen, Jarno Mäkinen
Tampere University of Technology
P.O. Box 553
FIN-33101 Tampere, Finland
{ari.visa, jarmo.toivonen, tomi.vesanen, jarno.makinen}@cs.tut.fi

Barbro Back
Åbo Akademi University
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland
barbro.back@abo.fi

Hannu Vanharanta
Pori School of Technology and Economics
P.O. Box 300, FIN-28101 Pori, Finland
hannu.vanharanta@pori.tut.fi

Abstract

We present two variations of a prototype based text matching methodology used in the Routing Sub-Task of TREC 2002 Filtering Track. The methodology examines text on the word level. It is based on word coding and examines the distributions of these codes using document histograms.

This research is supported by TEKES, the National Technology Agency of Finland (grant number 40943/99). The support is gratefully acknowledged.

1 Introduction

A common approach to topic detection and tracking is the usage of keywords, especially, in context of Dewey Decimal Classification [2, 1] that is used in United States to classify books. The approach is based on assumption that keywords given by authors or indexers characterize the text well. This may be true, but then one neglects the accuracy. There are also many automatic indexing approaches. A more accurate method is to use all the words of a document and the frequency distribution of words, but the comparison of frequency distributions is a complicated task. Some theories say that the rare words in the word frequency histograms distinguish documents [5]. Traditionally, information retrieval has roughly been based on a fixed list of index terms [5, 3], or vector space models [9, 8]. The latter ones miss the information of co-occurrences of words. There are techniques that are capable of considering the co-occurrences of words, as latent semantic analysis [6] but they are computationally heavy.

Commonly in filtering, documents are preprocessed with tokenizers, stemmers and stopword lists. Using these methods the processing of the documents become more simple for document classification methods. Next step is to construct feature vectors for documents. The value of the feature is usually based on its significance in the document. Traditionally this is done by using term frequencies and inverse document frequencies. Last year results of TREC 2001 filtering track show that using Support Vector Machine (SVM) for classification can give good results for the routing tasks [7, 4].

In this paper, we present our methodology briefly and concentrate on tests of content-based topic classification, which is highly attractive in text mining. The evolution of the methodology has been earlier discussed in several publications [10, 12, 11]. In the second chapter the applied methodology is described. In the third chapter the experiment with the Reuters database and execution times are described. Finally, the methodology and the results are discussed.

2 Methodology

The methodology used in our runs examines now the documents on the word level. The runs were designed so that the basic principles were kept the same. On the detailed level variation in the methods was added in order to test the robustness of the basic ideas.

2.1 Filtering

The original text was first preprocessed, extra spaces and carriage returns were omitted, and single words were separated with single spaces. With the Reuters database, the preprocessing included selecting the allowed XML fields and removal of the XML tags. For the Visa1T11 run a stopword list was created. Words which were common to the most of the topics where chosen into the stopword list. If the word occurred at least in 75 different topic it was chosen to the list. These words were regarded meaningless to the topic identification. For the Visa2T11 run the text was stemmed with the Porter stemmer.

2.2 Word quantization in Visa1T11

The filtered text was translated into a suitable form for encoding purposes. The encoding of words is a wide subject and there are several approaches for doing it. The word can be recognized and replaced with a code. This approach is sensitive to new words. The succeeding words can be replaced with a code. This method is language sensitive. Each word can be analyzed character by character and based on the characters a key entry to a code table is calculated. This approach is sensitive to capital letters and conjugation if the code table is not arranged in a special way.

The last alternative was selected, because it is accurate and suitable for statistical analysis. A word w was transformed into a number in the following manner:

$$y = \sum_{i=0}^{L-1} k^i * c_{L-i} \tag{1}$$

where L is the length of the character string (the word), c_i is the ASCII value of a character within a word w, and k is a constant.

Example: word is "**c a t**".

$$y = k^2 * ASCII(c) + k * ASCII(a) + ASCII(t) \tag{2}$$

The encoding algorithm produces a different number for each different word, only the same word can have an equal number. After each word has been converted to a code number, we consider the distribution of the code numbers of the words.

The representation of word coded numbers was floating point number. Floating point numbers in our system use a radix of two. Mantissa can have values from $[0.5, 1[$. The representation of floating point number:

$$mantissa * 2^{exponent} \tag{3}$$

Our word coding gives only positive numbers so sign is always positive. The mantissa has information of the beginning of the word and the exponent has information about the length of the word. The quantization of the words uses the values of the mantissa and the exponent. The range of the mantissa is divided to N equal size classes. The exponent is divided to size M classes. The mantissa class number and the exponent class number are used in the calculation of the word class number. Possible number of the mantissa classes of the word varies from 1 to N. The actual word class number is calculated in the following manner:

$$\begin{aligned} word\ class\ number &= \lfloor n \rfloor + (N * \lfloor m \rfloor) \\ n &= (y_{mantissa} - 0.5) * N * 2 \\ m &= \frac{y_{exponent}}{M}, \end{aligned} \tag{4}$$

where n is the mantissa class number of the word and m is the exponent class number of the word. N is the quantization accuracy of the mantissa, M is the quantization step of the exponent and y is the word coded to floating point number with formula 1.

Following example shows how the word coding and word class number generation is done to the word "trec". First the word is converted with word coding formula 1 to a number. Number is represented in floating point format where the radix is two. With formula 4 the word number is converted to a word class number, now N is 35000, M is 24, and k is 256.

$$\begin{aligned} y &= k^3 * ASCII(t) + k^2 * ASCII(r) + k^1 * ASCII(e) + k^0 * ASCII(c) \\ &= 0.909741090144962 * 2^{31} \end{aligned}$$

$$\begin{aligned} word\ class\ number &= \lfloor (y_{mantissa} - 0.5) * N * 2 \rfloor + (N * \lfloor y_{exponent}/M \rfloor) \\ &= 28681 + (35000 * 1) \\ &= 63681 \end{aligned} \tag{5}$$

2.3 Word quantization in Visa2T11

In the Visa2T11 run the word coding is a variation of the Visa1T11 word coding. Now, the alphabet of the training data is first determined from filtered and Porter stemmed training documents. The letters are put into order of their frequencies in the training data. The most frequent letter gets letter code 1, the second 2, and so on. If the letter does not appear in the training documents, it is given letter code 0. These letter codes are now used in formula 1 to replace the ASCII values.

Next, all the words of the training data are converted to word codes and their frequencies are counted. The word-frequency list is sorted according to the word code number. The word codes are classified to C classes using a simple classification scheme. The biggest gap between two succeeding word codes is first found and a class boundary is put between them. Then the sum of frequencies of words in the two new classes are counted. The class with most words is divided into two classes where the gap between two succeeding word codes is the biggest. This method is repeated until there are C classes. The class boundary information and the word codes are used in creating class numbers for the words of the documents.

2.4 Test document to histogram

When examining a single test document, we create a histogram of the word code numbers of the document. The filtered text from a test document is encoded word by word. Each word number is quantized using the word quantization method of the run. The quantization value is determined, an accumulator corresponding to the value is increased, and thus a word histogram A_w is created. The histogram A_w is finally normalized by the length of the histogram vector. The process of converting a document to a histogram is illustrated in Fig. 1. The histogram contains information about the words of the document in a numerical form. This histogram is used in the TREC Routing process to find the best topic for each test document. The diffence or distance between a single test document histogram and the histogram representing the topic can be calculated using different metrics. Among the most simple and effective metrics there are the Euclidean distance and the cosine distance.

With the histograms derived from all the documents in the test database we can compare and analyze the text of the single documents on the word level against the relevant texts of each topic. Note, that it is not necessary to have any prior knowledge of the actual text documents to use these methods. No linguistic methods, other than the Porter stemming, are used in the process.

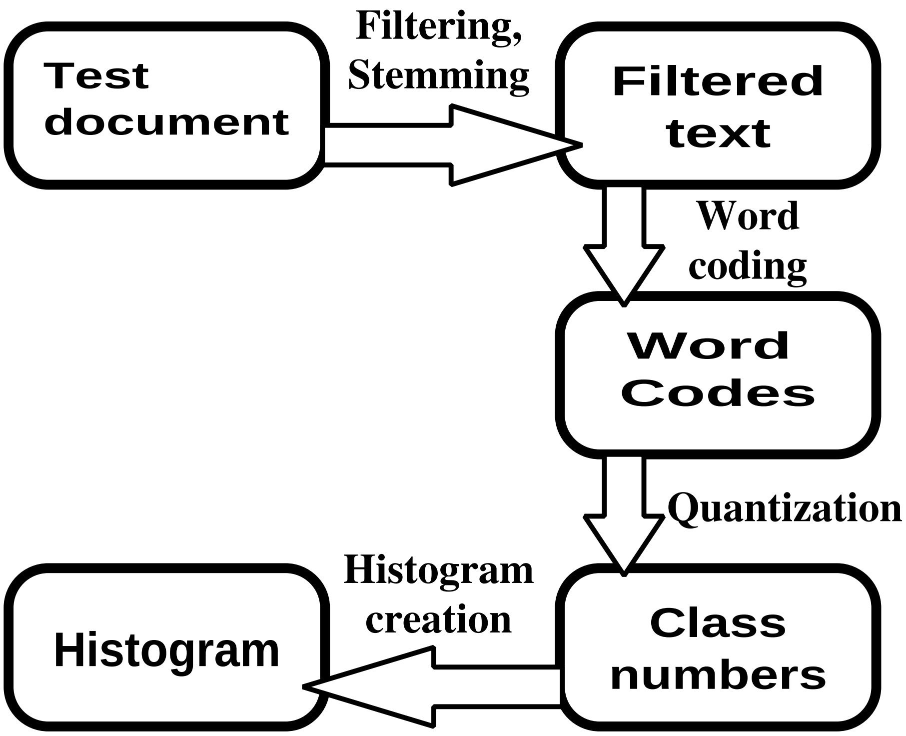

Figure 1: Process of converting document to histogram.

3 Runs with Reuters database

All the relevant documents to a certain topic from the training data were concatenated to one topic document. This document consists of all given relevant text documents classified to that topic. This document was used to define the topic. The information of irrelevant documents to the topic were not used in runs. Every topic document was converted to a normalized topic histogram. All test documents were also transformed to individual histograms and normalized to vector length one. In the two runs we used the methods described in sections 2.2 and 2.3.

Every test document histogram was compared with every topic histogram. The distance metric used in run Visa1T11 was the Euclidean distance. In run Visa2T11 the used distance metric was cosine distance. A topic best-match file was created for each topic. Each test document's four best matching (smallest distance) topics were determined. For these four topics, the ID number of the test document and its distance to the topic were put in the best-match file. From these files the top 1000 documents with the closest distance to the topic were selected for the result file. Our methods gave results which where close to the average level of all participating methods. Visa2T11 gave slightly better results than Visa1T11.

3.1 Execution times

The applied methodology is very fast even with a database as large as the Reuters database. In table 1 we present the execution times we calculated for the two runs. Making histograms execution time consists of creating the word histograms for the test documents. The comparing execution times are the times that it took to compare the test histograms with the topic histograms and to find the four closest topics for each test histogram.

Table 1: Execution times rounded up to the nearest hour.

	Making histograms	**Comparing**	**Altogether**
Visa1T11	1 h	6 h	7 h
Visa2T11	6 h	7 h	13 h

The computer used in the experiments was a PC with a Intel® 550 MHz Pentium® III processor and 128 Mb of memory. The operating system was Slackware Linux 7.0.0.

4 Discussion on results

There were some general difficulties when using the methodology on the Reuters database. The selection of documents for the given training set turned out to be disadvantageous. Firstly, it seemed that the set was too unevenly distributed in topics for our methodology. When some topics have under ten relevant documents and some hundreds, statistical methods are in trouble. There is not enough information in just few short relevant documents for this type of methods to be successful. Uneven division in topics also lead to give more weight to topics that have more relevant documents.

Secondly, because the training set was from a period of two months, the vocabulary in the relevant documents does not vary enough. The type of methodology we used requires a good set of representative word samples from the whole database. The training set vocabulary was restricted in the sense of yearly cycle, to two months in autumn of 1996. This type of difficulties are, on the other hand, very common in real life tasks.

Also, we had difficulties with the topics 151-200. Our methodology was not doing well in finding relevant test documents for these topics. This was perhaps partly due to our decision of emphasizing accuracy more than generalization. It may also be due to the nature of the artificial topic construction process.

Our runs were designed so that only a basic form of the methodology was used. The methods used are very fast and it seems that we are improving with the accuracy of the methodology. Visa2T11 had 10000 different classes for the words whereas Visa1T11 had about 6000. There was no training for the classification of words in Visa1T11. Because of smaller number of word classes Visa1T11 had one hour faster comparing time. The drawback was slightly poorer results. One interesting issue in advancing even more is how to use the information of the non-relevant documents for a topic to improve the process. Non-relevant documents seemed to be special cases of relevant documents topics. Our methodology can notll use the information of non-relevant document, because only few words can make distinction between relevant and non-relevant document. In future, this could maybe be achieved by giving negative weight to those kind of words.

References

[1] M. Dewey. *A Classification and subject index for cataloguing and arranging the books and pamphlets of a library*. Case, Lockwood & Brainard Co., Amherst, MA, USA, 1876.

[2] M. Dewey. Catalogs and Cataloguing: A Decimal Classification and Subject Index. In *U.S. Bureau of Education Special Report on Public Libraries Part I*, pages 623–648. U.S.G.P.O., Washington DC, USA, 1876.

[3] T. Lahtinen. *Automatic indexing: an approach using an index term corpus and combining linguistic and statistical methods*. PhD thesis, Department of General Linguistics, University of Helsinki, Finland, 2000.

[4] D. D. Lewis. Applying Support Vector Machines to the TREC-2001 Batch and Routing Tasks. In E. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, NIST Special Publication 500-250, pages 286–292, Gaithersburg, Maryland, USA, November 13–16 2001. Department of Commerce, National Institute of Standards and Technology (NIST).

[5] C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, 1999.

[6] D. W. Oard and G. Marchionini. A conceptual framework for text filtering. Technical Report CS-TR3643, University of Maryland, May 1996.

[7] S. Robertson and I. Soboroff. The TREC 2001 Filtering Track Report. In E. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, NIST Special Publication 500-250, pages 26–37, Gaithersburg, Maryland, USA, November 13–16 2001. Department of Commerce, National Institute of Standards and Technology (NIST).

[8] G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.

[9] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[10] J. Toivonen, A. Visa, T. Vesanen, B. Back, and H. Vanharanta. Validation of Text Clustering Based on Document Contents. In P. Perner, editor, *Proceedings of MLDM 2001, the Second International Workshop on Machine Learning and Data Mining in Pattern Recognition*, number 2123 in Lecture Notes in Artificial Intelligence, pages 184–195, Leipzig, Germany, July 25–27 2001. Springer–Verlag.

[11] A. Visa, J. Toivonen, H. Vanharanta, and B. Back. Contents Matching Defined by Prototypes – Methodology Verification with Books of the Bible. *Journal of Management Information Systems*, 18(4):87–100, 2002.

[12] A. Visa, J. Toivonen, T. Vesanen, and J. Mäkinen. Tampere University of Technology at TREC 2001. In E. Voorhees and D. Harman, editors, *Proceedings of the Tenth Text REtrieval Conference (TREC 2001)*, NIST Special Publication 500-250, pages 495–501, Gaithersburg, Maryland, USA, November 13–16 2001. Department of Commerce, National Institute of Standards and Technology (NIST).