Toward Text Understanding – Comparison of Text Documents by Sentence Map

Ari Visa, Jarmo Toivonen, Barbro Back* and Hannu Vanharanta†
Lappeenranta University of Technology
P.O. Box 20, FIN-53851 Lappeenranta Finland
email: {Ari.Visa, Jarmo.Toivonen}@lut.fi
*Åbo Akademi University
Lemminkäisenkatu 14 A, FIN-20520 Turku Finland
email: Barbro.Back@abo.fi
†Pori School of Technology and Economics
Tampere University of Technology
Bredantie 28-30 C 6, FIN-02700 Kauniainen Finland
email: Hannu.Vanharanta@pori.tut.fi

ABSTRACT: The automatic classification and analysis of documents according to their content is a topic of interest to people working in many fields of science and research, such as accounting, financing, marketing, engineering, and law. Neural networks offer an adaptive tool for this purpose. The field-specific vocabularies are often large and high accuracy in word separation is usually a requirement (for example, words $cost$, $Cost$ and $costs$ have a different meaning). One often encountered problem is that the vocabularies are open, meaning that new words may appear. This paper suggests a new adaptive approach to the problem of automatic document classification and analysis. The approach is based on analysing the relations between succeeding words in sentences. The methodology used is based on smart encoding, on Self-Organizing Maps, and on document histograms. The results are promising, and they provide new ways in which to analyse text documents in different contexts.

KEYWORDS: data mining, neural networks, text classification, self-organizing maps

INTRODUCTION

The pressure toward more natural user-interfaces has made the discipline of natural language understanding, together with the specific sub-discipline of automatic text understanding, both subjects of increasing importance. This is due to the Internet, to office automation, and to the plummeting cost of computation. The amount of available text documents is vast, and it is increasing continuously.

It has been common to approach these challenges through the use of automata theories, grammars, or language theories. There exist many successful commercial software applications, such as spelling checkers and language translation programs. For some applications, rule-based systems have also proved useful. For classification purposes, it has been common to use a limited number of given keywords. Neural network methods offer a different, associative, approach to the problem. In particular, this is true of those neural networks that use competitive learning, for instance Self-Organizing Maps (SOM) [2, 5]. The only requirement is that there be a large quantity of text documents available. The main principles for the use of SOM were published by Ritter and Kohonen [9] in 1989. Today, ten years later, there exist several SOM-based applications for the browsing of documents and for information retrieval and data mining [7, 10, 11, 8].

One problem in the use of SOM lies in how to construct numerical representations of documents so that they contain the relevant information about the contents of the document. One method in text retrieval is to encode a document as the histogram of its words. In such a format, the information regarding relative word order in the document is lost but efficiency of representation is gained.

In large document collections the vocabularies may become prohibitively large. In WEBSOM [1, 4], word category histograms are used. The words are clustered with the aid of so-called “self-organizing semantic maps” [9] that use the statistics regarding the textual contexts of words to provide information on their relations. In this way the size of the

word histograms can be reduced. The semantic similarity of the words is expressed in terms of the proximity of the word clusters to each other on the semantic map. The proximity can be taken into account in the encoding of the documents. This approach has proved suitable for browsing purposes but, in cases where the vocabulary is large and almost fixed, the achieved accuracy is insufficient.

In this article the above-indicated problem is first discussed and formulated, the proposed method is briefly described, some preliminary results are presented and finally the quality of the proposed method is discussed.

THEORETICAL BASIS

Assume a set of words W, a subset of words N (a dictionary) and $N \subset W$. A word w will usually belong to the dictionary but this is not necessarily the case, i.e. $\{\exists w | w \subset W \wedge w \not\subset N\}$. This formulation makes it difficult to use grammars or rules. To achieve adaptability to different vocabularies and to solve the problem that new words may appear, the dictionary N is built up through unsupervised learning using a clustering process. The only practical requirement is that there are large amounts of representative text documents available. By choosing the unsupervised learning method carefully it is easy to define relations between successive words w_i , w_{i+1}, w_{i+2}, w_{i+3}, ... All these relationships can then be used to classify a text document.

PROPOSED SOLUTION

The original text is first filtered; this means that compound words are treated in a certain way, numbers are treated in a certain way, and so on. The filtered text must then be translated into a suitable form for clustering purposes. This is done by encoding. A word w is transformed into a number in the following manner:

$$y = \sum_{i=0}^{L-1} 2^{4i} * c_{L-i} \tag{1}$$

where L is the length of the character string (the word), c_i is a character within a word, and k is a constant. The word w is now given a value, a word vector $\mathbf{Y}$, by a tabulated function. The table has a size of $N * M$, where M is the length of the word vector. Now

$$\mathbf{Y} = f(y) \bmod P \tag{2}$$

where P is a suitable prime and $P = N$. The table is Gray coded and the actual code is produced from binary code in an iterating manner. Note that $x < N$

$$\begin{aligned} y_1 &= x_1 \\ y_2 &= x_1 \otimes x_2 \\ y_3 &= x_2 \otimes x_3 \\ &\cdot \\ &\cdot \\ &\cdot \\ y_N &= x_{N-1} \otimes x_N \end{aligned} \tag{3}$$

where x represents the original binary code, y represents the produced binary code, and $\otimes$ is a logical exclusive or operation. The word vectors are clustered using, for instance, a Self-Organizing Feature Map [2]

$$\begin{aligned} \mathbf{n_i}(t+1) &= \mathbf{n_i}(t) + \alpha(t) * [\mathbf{Y}(t) - \mathbf{n_i}(t)], \text{ for } i \in N_c \\ \mathbf{n_i}(t+1) &= \mathbf{n_i}(t), \text{ for } i \notin N_c \end{aligned} \tag{4}$$

where $\alpha(t)$ is a decreasing function with the following properties:

$$\sum_{t=-\infty}^{\infty} \alpha(t) = \infty, \qquad \sum_{t=-\infty}^{\infty} \alpha(t)^2 < \infty \tag{5}$$

The neighbourhood N_c should also have a decreasing behaviour as a function of time t. Some other algorithms are also possible. As a result of the clustering process a word map of N elements is created. The process of creating a word map is

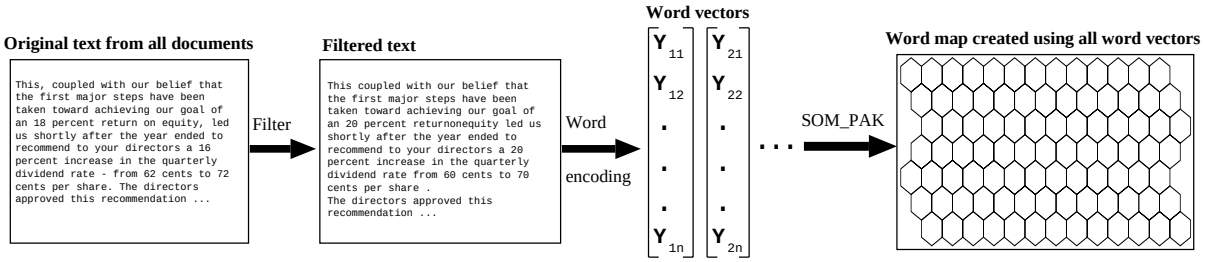

Figure 1: Creating the word map.

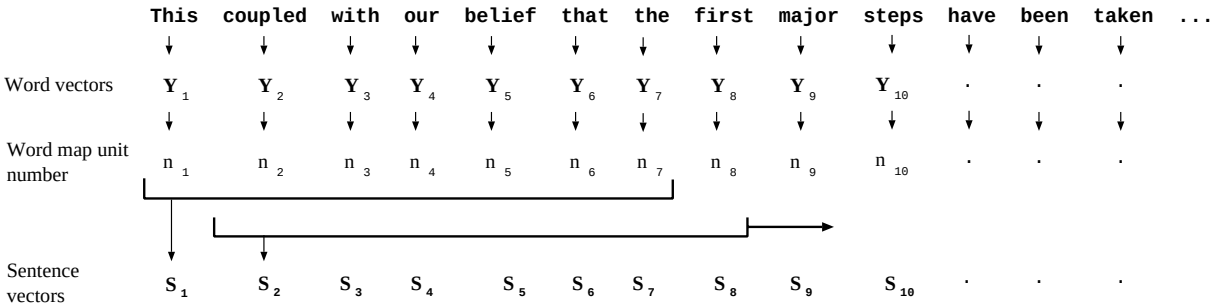

Figure 2: Creating the sentence vectors from the encoded words.

illustrated in Figure 1. The filtered text is encoded word by word. Next, a small neighbourhood of encoded words is taken as input to a second self-organizing process. The small neighbourhood glides step by step over the sentence and produces inputs to the SOM. These inputs produce a sentence map. The process is illustrated in Figure 2. All the documents used for teaching the SOM are processed in similar manner. In other words, SOM is used twice, first to produce a word map and then to produce a sentence map.

The text documents are first encoded word by word comparing with the word map. The classified words are replaced with the corresponding word unit numbers. The unit numbers are concatenated together to a sentence vector. Each sentence vector is compared with a sentence map. The best match is determined by

$$||\mathbf{Y} - \mathbf{n_c}|| = min_i||\mathbf{Y} - \mathbf{n_i}|| \tag{6}$$

and an accumulator corresponding to the best matching element $\mathbf{n_c}$ on the sentence map is increased. The generated histogram A consisting of N bins is then normalized by the sentence count of the documents. The whole text classification process is illustrated in Figure 3. Now it is possible to compare and analyse two or more text documents, as can be seen in Figure 4. Note that it is not necessary to know anything about the actual text documents to compare and to classify the documents. For classification purposes vector quantization or Learning Vector Quantization (LVQ) [3] is used.

EXPERIMENTS

The solution proposed above has been programmed as a prototype software application. The principles of this application have been verified by means of experiments. The aims of the experiments are twofold: firstly, to demonstrate the possibilities of a sentence map and, secondly, to demonstrate the functionality of the application in a real case; a sentence map analysis of an annual report.

The first experiment demonstrates that the method is capable of separating short sentences. The results are represented as a labeled sentence map which, in this experiment, is of size 20*15. The size of the wordmap used here is 52*40. Figure 5 shows a part of the sentence map with sentences consisting of three successive words. It is possible to achieve the desired accuracy by using the proposed method. In the second experiment, the software is applied on real data; two annual reports of the US company International Paper, from the years 1987 and 1988. The histograms A are represented in Figure 6. As can be seen, the histograms differ from each other. Levenshtein metrics [6] is used in histogram comparison. As an example 15 annual reports are compared and the results are shown in Table1. As can be seen the succeeding annual reports usually resemble each other but there are also some exceptions.

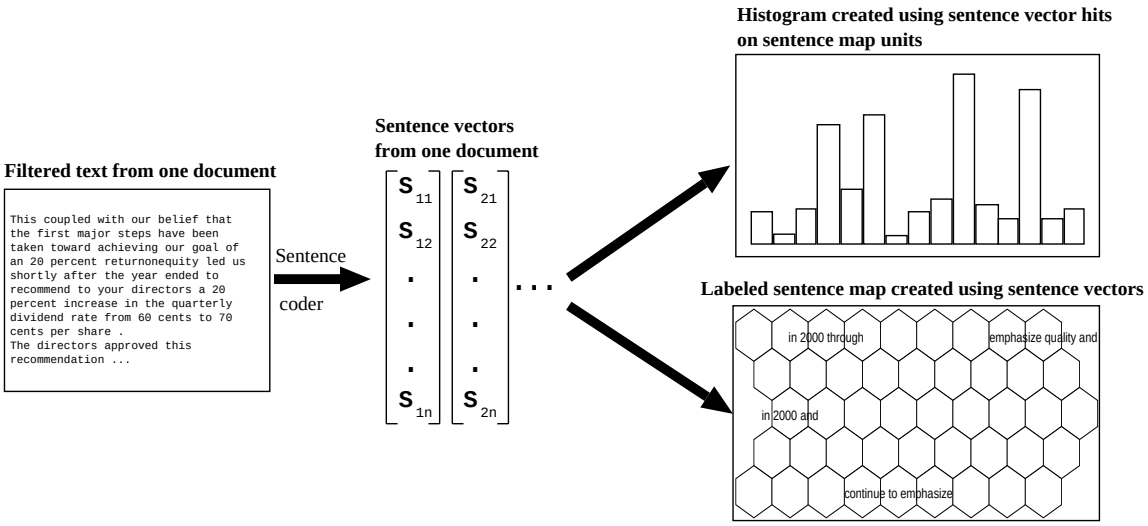

Figure 3: The text classification process.

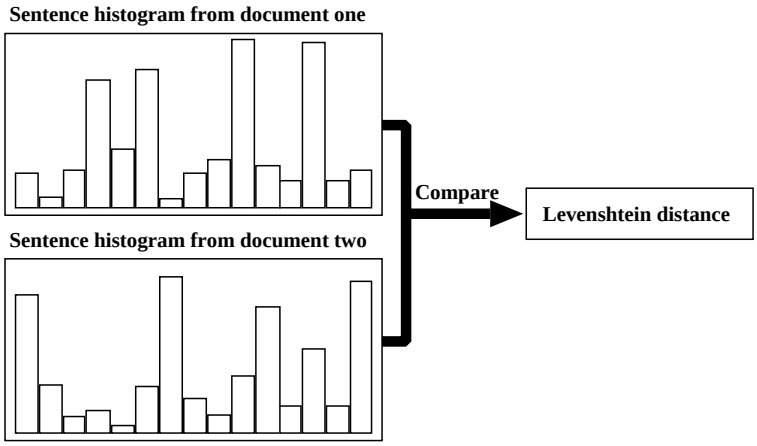

Figure 4: Comparing and analysing documents based on the extracted sentence histograms.

DISCUSSION

Conventionally, text documents have been characterized using certain keywords. Also in our approach the characterization is based on a considerable number of keywords, but now the keywords are first learnt in an unsupervised manner from the given documents. The learning takes place using neural network methods. This makes it possible to adapt the methodology to the vocabularies of different user groups. The unsupervised learning approach requires that there be representative text documents available. The size of the vocabulary N is chosen by the needs of the user group. However, it is not necessary that the size of the vocabulary N in the proposed approach be as large as in conventional histogram-based classification methods. The comparison and classification itself is based on histograms of co-occurrence of successive words. Our approach thus differs from other published methods in several crucial respects. The main differences between the WEBSOM method and the proposed method is that the proposed method is simpler and that no distance-based weighting is used in the encoding of the neighbourhood.

The fact that there may appear words in the text that are not included in the dictionary N is a big advantage in favor of the proposed approach. The encoding method, the unsupervised way of inputting words into the dictionary, and the classification based on distance metrics all make it possible to consider new words easily.

The choice of word encoding and clustering methods is crucial to the success of the whole classification process; the methods must have a high degree of compatibility. The encoding method chosen depends on the initial problem; it should be chosen for each specific case. Also the choice of the size of the neighbourhood is crucial. The larger the neighbourhood the better the specificity but the bigger the problems. For instance, short sentences are problematic. The amount of

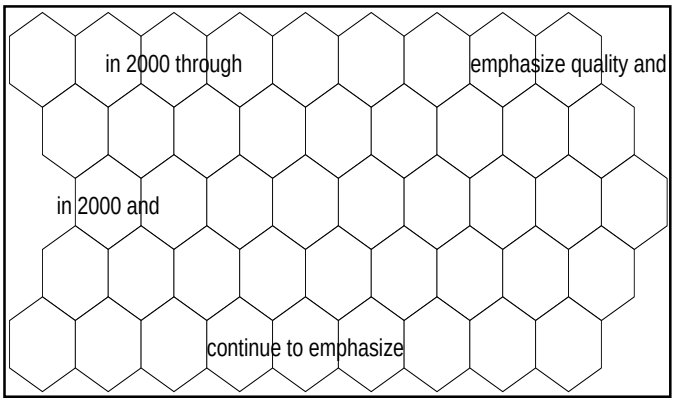

Figure 5: A demonstration of the possibilities of the sentence map.

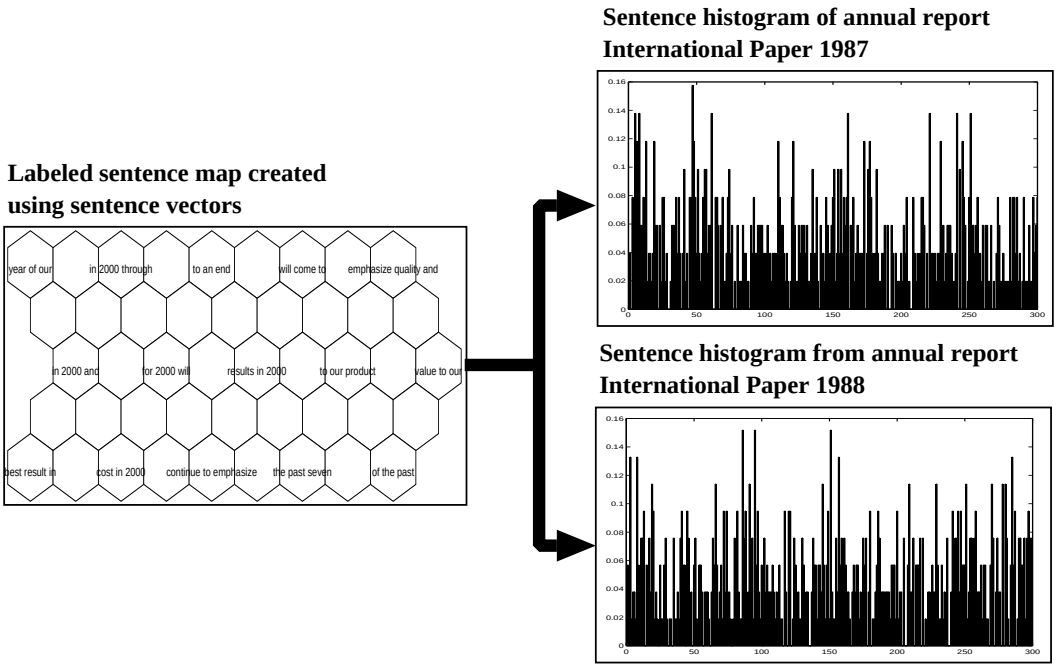

Figure 6: A demonstration on two consecutive annual reports of the same company.

available text documents might also be a problem.

The histogram approach makes it possible to gain accuracy and speed in comparison. However, in conventional methods the information related to successive words is usually lost. This drawback is avoided in the proposed approach in which histograms of sentence vectors are gathered. Sentence vectors consist of encoded successive words.

The proposed method is very promising. The method has been tested on several hundred annual reports with success. At present we are using the proposed method to analyse the characteristics of the clusters and the behaviour of the time series of the tested annual reports.

ACKNOWLEDGEMENTS

The authors thank Mr. Mikko Irjala and Ms. Piia Ruokonen for the programming help.
The financial support of TEKES (grant number 40887/97) is gratefully acknowledged.

	kc85	kc86	kc87	kc88	kc89	mb85	mb86	mb87	mb88	mb89	sco85	sco86	sco87	sco88	sco89
kc85	0	0.2231	0.2136	0.2047	0.2134	0.2096	0.2175	0.2319	0.2360	0.2325	0.2095	0.2001	0.2097	0.1881	0.2036
kc86	0.2231	0	0.2172	0.2089	0.2311	0.2098	0.2030	0.2262	0.2392	0.2226	0.2044	0.2071	0.2336	0.2039	0.2059
kc87	0.2136	0.2172	0	0.1929	0.2070	0.1987	0.2128	0.2165	0.2252	0.1975	0.1945	0.1763	0.2036	0.2041	0.1879
kc88	0.2047	0.2089	0.1929	0	0.1365	0.1241	0.1101	0.1312	0.1039	0.1039	0.1360	0.1328	0.1621	0.1187	0.1194
kc89	0.2134	0.2311	0.2070	0.1365	0	0.1392	0.1338	0.1329	0.1329	0.1304	0.1549	0.1497	0.1708	0.1508	0.1529
mb85	0.2096	0.2098	0.1987	0.1241	0.1392	0	0.1064	0.1241	0.1071	0.1020	0.1336	0.1325	0.1593	0.1330	0.1239
mb86	0.2175	0.2030	0.2128	0.1101	0.1338	0.1064	0	0.1132	0.0854	0.0855	0.1237	0.1321	0.1610	0.1253	0.1218
mb87	0.2319	0.2262	0.2165	0.1312	0.1329	0.1241	0.1132	0	0.1098	0.1022	0.1357	0.1454	0.1582	0.1318	0.1342
mb88	0.2360	0.2392	0.2252	0.1039	0.1329	0.1071	0.0854	0.1098	0	0.0928	0.1393	0.1311	0.1567	0.1119	0.1385
mb89	0.2325	0.2226	0.1975	0.1039	0.1304	0.1020	0.0855	0.1022	0.0928	0	0.1305	0.1301	0.1566	0.1249	0.1197
sco85	0.2095	0.2044	0.1945	0.1360	0.1549	0.1336	0.1237	0.1357	0.1393	0.1305	0	0.1381	0.1735	0.1545	0.1504
sco86	0.2001	0.2071	0.1763	0.1328	0.1497	0.1325	0.1321	0.1454	0.1311	0.1301	0.1381	0	0.1779	0.1498	0.1415
sco87	0.2097	0.2336	0.2036	0.1621	0.1708	0.1593	0.1610	0.1582	0.1567	0.1566	0.1735	0.1779	0	0.1550	0.1656
sco88	0.1881	0.2039	0.2041	0.1187	0.1508	0.1330	0.1253	0.1318	0.1119	0.1249	0.1545	0.1498	0.1550	0	0.1334
sco89	0.2036	0.2059	0.1879	0.1194	0.1529	0.1239	0.1218	0.1342	0.1385	0.1197	0.1504	0.1415	0.1656	0.1334	0

kc=Kimberly-Clark, mb=MacMillan-Bloedel, sco=Scott Paper USA

Table 1: A confusion matrix showing the Levenshtein distance between 15 annual reports.

REFERENCES

[1] Timo Honkela, Samuel Kaski, Krista Lagus, and Teuvo Kohonen. Newsgroup Exploration with WEBSOM method and Browsing Interface. Technical Report A32, Helsinki University of Technology, 1996.

[2] T. Kohonen. Self-Organized formation of topologically correct feature maps. *Biological Cybernetics*, 43:59–69, 1982.

[3] T. Kohonen. Learning Vector Quantization for Pattern Recognition. Technical Report TKK-F-A601, Helsinki University of Technology, Department of Technical Physics, Laboratory of Computer and Information Science, 1986.

[4] T. Kohonen, S. Kaski, K. Lagus, and T. Honkela. Very Large Two-Level SOM for Browsing of Newsgroups. In *Proc. of ICANN'96 International Conference on Artificial Neural Networks*, pages 269–274. Springer, 1996.

[5] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Series in Information Sciences*. Springer-Verlag, Heidelberg, 2. edition, 1997.

[6] Vladimir I. Levenshtein. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 8(10):707–710, 1966.

[7] X. Lin, D. Soergel, and G. Marchionini. A Self-Organizing Semantic Map for Information Retrieval. In *Proc. of 14th Ann. Int. ACM/SIGIR Conf. on R&D in Information Retrieval*, pages 262–269, 1991.

[8] D. Merkl, A. Tjoa, and G. Kappel. A Self-Organizing Map that Learns the Semantic Similarity of Reusable Software Components. In *Proc. of ACNN'94, 5th Australian Conf. on Neural Networks*, pages 13–16, 1994.

[9] H. Ritter and T. Kohonen. Self-Organizing Semantic Maps. *Biological Cybernetics*, 61(4):241–254, 1989.

[10] J. C. Scholtes. Unsupervised learning and the information retrieval problem. In *Proc. of IJCNN'91, Int. Joint Conf. on Neural Networks*, volume I, pages 95–100. IEEE Service Center, 1991.

[11] Alfred Ultsch. Knowledge Acquisition with Self-Organizing Neural Networks. In I. Aleksander and J. Taylor, editors, *Artificial Neural Networks, 2*, volume I, pages 735–738. North-Holland, 1992.