Chapter 10

Self-organization of very large document collections

Teuvo Kohonen, Samuel Kaski, Krista Lagus, Jarkko Salojärvi, Jukka Honkela, Vesa Paatero, Antti Saarela

Text mining systems are developed to aid the users in satisfying their information needs, which may vary from searching answers to well-specified questions to learning more of a scientific discipline. The major tasks of web mining are *searching*, *browsing*, and *visualization*. Searching is best suited for answering specific questions of a well-informed user. Browsing and visualization, on the other hand, are beneficial especially when the information need is more general, or the topic area is new to the user [6]. The SOM, applied to organizing very large document collections, can aid in all the three tasks.

Statistical models of documents

In the vast majority of SOM applications, the input data constitute high-dimensional real *feature vectors*. In the SOMs that form similarity graphs of *text documents*, the feature sets that describe collections of words in the documents should be expressible as real vectors, too. The feature sets can simply be weighted histograms of the words, but usually some dimensionality reduction of the histograms is carried out.

The primitive vector space model

In the basic *vector space model* [1] the stored documents are represented as real vectors in which each component corresponds to the frequency of occurrence of a particular word in the document: the model or document vector can be viewed as a weighted word histogram. For the weighting of a word according to its importance one can use the Shannon entropy over document classes, or the inverse of the number of the documents in which the word occurs ("inverse document frequency"). The main problem of the vector space model is the large vocabulary in any sizable collection of free-text documents, which means a vast dimensionality of the model vectors.

Latent semantic indexing (LSI)

In an attempt to reduce the dimensionality of the document vectors without essentially losing information contained in the full vocabulary, one often first forms a matrix in which each column corresponds to the word histogram of a document, and there is one column for each document. After that the space spanned by the column vectors is decomposed into an ordered set of factors by a matrix-computation method called the singular-value decomposition (SVD). The decomposition has the property that the last factors have minimal influence on the matrix. When they are omitted the document vector formed from the histogram of the remaining factors has then a much smaller dimensionality while as much as possible is retained of the original histograms. This method is called *latent semantic indexing (LSI)* [2].

Randomly projected histograms

We have shown that the dimensionality of the document vectors can be reduced radically by a much simpler and computationally lighter method than the LSI, the random projection method [3], without essentially losing the power of discrimination between the documents. Consider the original document vector (weighted histogram) $\mathbf{n}_i \in \mathbb{R}^n$ and a rectangular random matrix $\mathbf{R}$, the elements in each column of which are assumed to be normally distributed. Let us form the document vectors as the *projections* $\mathbf{x}_i \in \mathbb{R}^m$, where $m \ll n$:

$$\mathbf{x}_i = \mathbf{R}\mathbf{n}_i \; . \tag{10.1}$$

It can be shown that for large m, the similarity relations between arbitrary pairs of projection vectors $(\mathbf{x}_i, \mathbf{x}_j)$ are very good approximations of the corresponding relations between the original document vectors $(\mathbf{n}_i, \mathbf{n}_j)$, and the computing load of the projections is reasonable. The variance of the error is inversely proportional to m, and in our experiments values of m exeeding 100 have yielded practically as good results as the original vectors (see [3] for theoretical considerations and [5] for comparison results). On the other hand, with the radically decreased dimensionality of the document vectors, the time needed to classify a document is radically decreased.

Construction of the random projections by pointers

There exists a special method for forming projections that gives as good results as the random projections discussed above but is computationally much more efficient for sparse input vectors. This method is used in our latest experiments, described e.g. in [5].

Construction of the document map

A brief overview of the computing phases is given in the following.

Preprocessing. From the raw text, nontextual and otherwise nonrelevant information (punctuation marks, articles and other stopwords, message headers, URLs, email addresses, signatures, images, and numbers) was removed. The most common words, and words occuring rarely (e.g., less than 50 times in the corpus) were also discarded.

The variation caused by different word forms and inflections has been reduced in our later experiments by using a *stemmer* for both Finnish and English text.

Formation of statistical models. To reduce the dimensionality of the models, we have used both randomly projected word category histograms (see, e.g., [5]) and randomly projected word histograms (or their shortcut computation by pointers), weighted by the Shannon entropy or the "inverse document frequency" (idf).

Formation of the document map. The document maps were formed automatically by the SOM algorithm, for which the statistical models of documents were used as input. The size of the SOM was determined so that on the average 7 articles were mapped onto each grid point; this figure was mainly determined for the convenience of browsing.

The speed of computation, especially of large SOMs can be increased by several methods. In our latest experiments we have used the Batch Map algorithm [5], and furthermore, the winner search was accelerated in several ways. The search was started in the neighborhood of corresponding winners at the previous cycle of iteration. The distance computation, or in this case actually computation of inner products, can be speeded up even further by neglecting the components having the value zero; a large proportion of the components are zeroes since the input vectors are still sparse after the dimensionality has been reduced by the random projection with pointers.

The size (number of grid nodes) of the maps was increased stepwise by estimating a larger map based on a smaller one. After each increase the winners for all data vectors can be found quickly based on the estimation formula used for increasing the map. If necessary, the winner search can be speeded up even further by restricting the search to the area of the larger map that corresponds to the neighborhood of the winner on the smaller map.

For very large maps it may be necessary to reduce the amount of memory needed for storing the maps. We have represented the model vectors with reduced accuracy and used probabilistic updating to maintain numerical accuracy in adapting the model vectors.

To evaluate the performance of the speedup methods in constructing document maps we compared document maps computed using the shortcut methods with maps computed by the traditional SOM algorithm. As can be seen from Table 10.1, the quality of the resulting document maps is comparable, but the time needed for the shortcut methods is only about one tenth of that of the traditional algorithm.

Table 10.1: Comparison of the shortcut methods with the traditional SOM algorithm. The figures are averages from five test runs with different random matrices used in the encoding of the documents, and the error margins are standard deviations. The data was 13,742 abstracts from 21 patent classes, and the final maps consisted of 1,344 units. The time has been measured with a SGI O2000 computer without parallelization of any programs.

	Classification accuracy (%)	Quantization error	Time (s)
Traditional SOM	58.2 ± 0.2	0.799 ± 0.001	2550 ± 40
Shortcut methods	58.0 ± 0.2	0.798 ± 0.002	241 ± 3.5

For very large maps the difference in the computation times is even more marked than in Table 10.1, but can only be deduced from the computational complexities given in Table 10.2.

User interface. The document map is presented as a series of HTML pages and clickable images that enable exploration of the grid points: a mouse click on a grid point brings to view the links to documents residing in that grid point. The documents, stored in a database, can then be read by following the links. A large map can be first zoomed to view subsets of it more closely. For the largest maps we have used several zooming levels. To provide guidance in the exploration, an automatic method, described in [8], has been

Table 10.2: Computational complexity of the methods. Here N denotes the number of data samples, M the number of map units in the small map, and d the dimensionality of the input vectors. It has been assumed that the number of map units in the final map is chosen to be proportional to the number of data samples.

	Computational complexity
Traditional SOM	$\mathcal{O}(dN^2)$
Shortcut methods	$\mathcal{O}(dM^2) + \mathcal{O}(dN) + \mathcal{O}(N^2)$

utilized for selecting keywords to characterize map regions. The selected words have been marked on the map display.

Content-addressable search. The interface to the map has been provided with a form field into which the user can type a query in the form of a short "document." This query is preprocessed and a document vector is formed in the exactly same manner as for the stored documents. The resulting vector is then compared with the "models" of all SOM grid points, and the best-matching points are marked with circles on the map display: the better the match, the larger the circle. These locations provide good starting points for browsing.

If the document map is very large, the comparison between the document vector and all the model vectors is time-consuming. It is, however, possible to make rapid approximations by restricting the comparisons in such subspaces of the original space that best represent the (local) organization of the map.

Keyword search. If the user wants to find documents containing a single keyword or very few keywords, one can search the map using a more conventional *keyword search mode* which is provided as an alternative to the content addressable search. The keyword search is performed by accessing an index from each word to the map units where that word occurs.

Example: The largest published map

For the largest WEBSOM map made so far we selected a data base of 6,840,568 patent abstracts available in electronic form and written in English. These patents were granted by the U.S., European, and Japan patent offices and stored in two databases: the "First Page" database (1970-1997), and "Patent Abstracts of Japan" (1976-1997). The average length of the texts was 132 words. The size of the finally accepted vocabulary was 43,222 words. The words were weighted by the Shannon entropy computed from the distribution of the words into 21 subsections from the IPC6 patent classification.

The size of the SOM was 1,002,240 models (neurons), and the dimensionality of each model was 500. The representation for each document has been made by projecting the 43,222-dimensional word histogram randomly onto the final 500-dimensional space. Formation of the document map and the interface took altogether about 6 weeks with the newest speedup methods; searching occurs in a few seconds.

The accuracy of classifying a document into one of the 21 subsections was about 64 per cent. It must be noted that the categories are in fact overlapping: many patents belong to several different categories.

Figure 10.1 exemplifies a case of content-addressable search. The map and the collection can be explored using a WWW-browser. Further examples of document maps can be found at `//websom.hut.fi/websom/`.

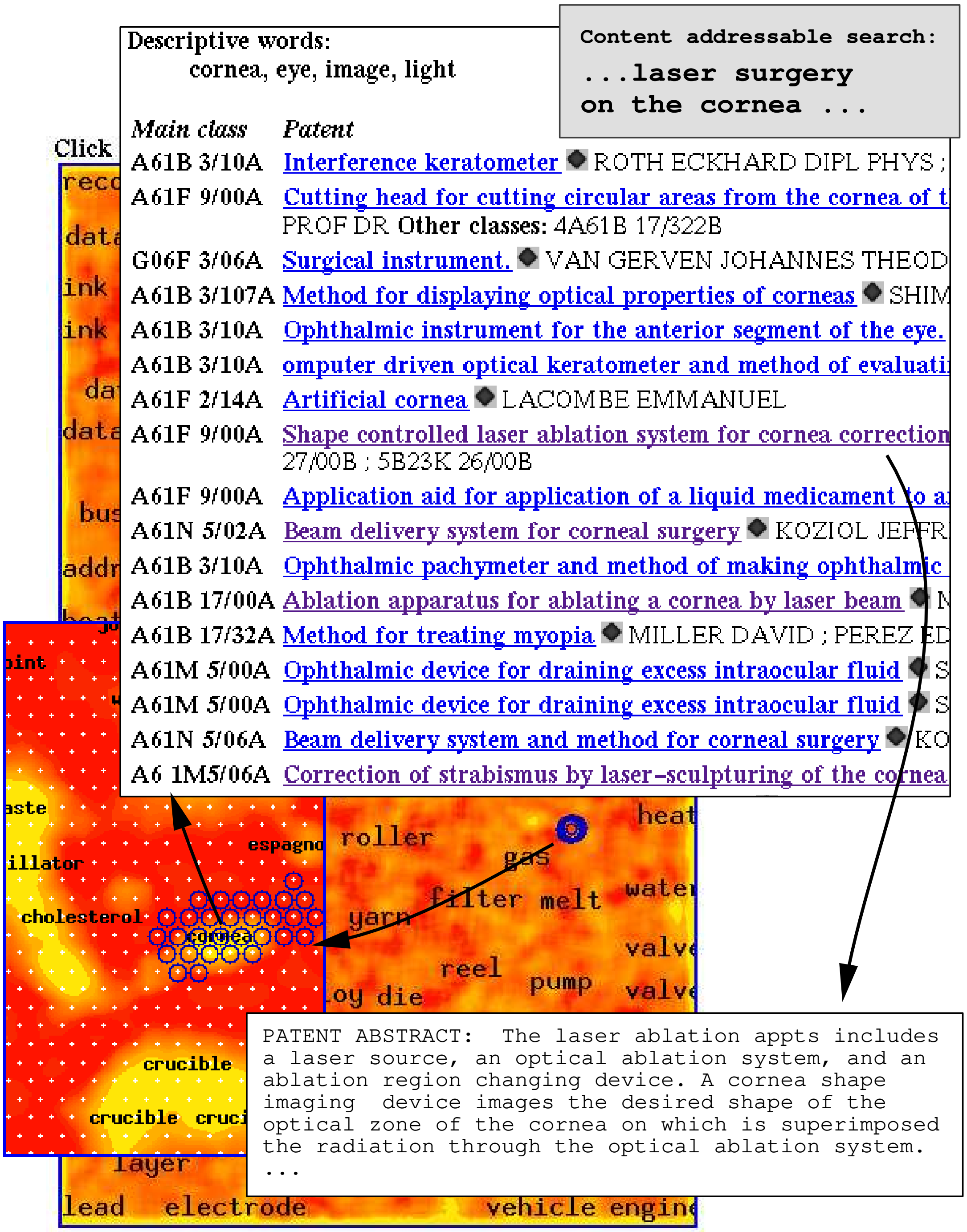

Figure 10.1: Content addressable search was utilized on a WEBSOM document map of nearly 7 million patent abstracts. The document map is depicted in the background, and the shades of gray correspond with document clusters. The search query was “laser surgery on the cornea” and the best-matching map locations are marked with circles. Zooming on the area reveals a small cluster of map units that contains patent abstracts mostly about the cornea of the eye, and of surgical operations on it. The words written on the document map have been selected automatically using the method described in [8].

Using the document maps for improving search results

Previously, it has been shown how the maps can be utilized for exploration of a large document collection with the help of a browsing interface and the visualized map display. However, as described in [7] and in the following, the document maps can also be applied for searching without the benefit of the visual interface. This allows comparison to systems that do not offer any visualization or exploration capabilities.

The search algorithm

Consider the typical search task where one wishes to locate the n best documents corresponding to a query, where n is a small number, say 50, and the documents are to be ordered based on goodness of match. The following algorithm is proposed for performing such searches using a document map:

1. Off-line organizatory phase: Cluster the documents on a WEBSOM document map with M map units. Find the best-matching unit on the map for each document. Represent the set of documents within a map unit by the model vector of that unit.

2. On-line search phase: For a given query,

    (a) *Pre-select:* Compare the query vector with the model vectors to select the best map units (based on minimum inner product distance from the query vector), and collect documents from those units until a total of K documents ($K >= n$) are obtained.

    (b) *Refine:* perform an exhaustive search among the K prospective documents and return the n best ones in the order of goodness. In the straightforward case the exhaustive search amounts to comparing the document vectors to the query vector, and measuring similarity by inner products. Alternatively, a more detailed search could be carried out.

The hierarchical search is computationally efficient: The full index of the document collection need not be held in main memory, nor is there a need to search through all the documents.

Experiment on text retrieval

The usefulness of the approach was studied on the well-known CISI reference collection for text retrieval. The CISI data set consists of 1460 documents and 76 queries. The collection lists for each query all the documents that are considered relevant to the query (41 on the average), as judged by a human. The average lengths of the documents and the queries were 115 and 51 words, respectively. The CISI collection is generally considered difficult for retrieval systems.

Creation of the document map. The documents were organized on a WEBSOM map of 150 units. The preprocessing was carried out as described above. The size of the vocabulary was 2136 words, small enough that no random projection was applied in document encoding.

Selecting a value for the K parameter. The set of queries was split randomly to two parts. The first half was used to select a value for the K parameter (the number of documents chosen at the pre-selection phase) and the second half for evaluating the results.

Measuring accuracy of retrieval. The accuracy was measured using *non-interpolated average precision over all relevant documents* (AP) [9]. AP is calculated for a single query by recording the precision each time a relevant document is retrieved, and taking an average of these (the precision of non-retrieved documents is taken to be zero). The results are further averaged over all queries. The measure not only takes into account the *number* of relevant documents retrieved, but also their *order* in the result set, rewarding systems for ranking relevant documents high in the returned lists of results.

Results. The results of a comparison between the proposed two-stage method, vector space model, and LSI are shown in Figure 10.2. The improvement obtained using docu-

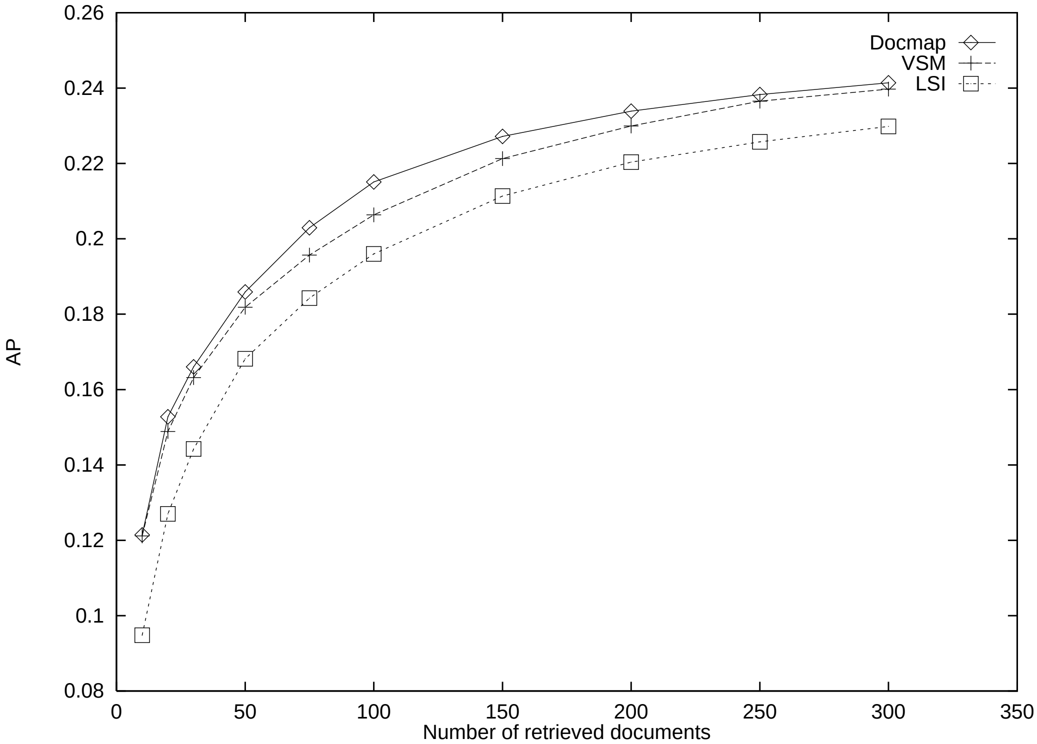

Figure 10.2: Comparison of methods. The average precision (AP) was calculated using the suggested document map method (Docmap), the vector space model (VSM) and the Latent Semantic Indexing (LSI) with various numbers of retrieved documents. The results are averages over the queries in the second half of the test set.

ment maps when compared to the vector space model was found to be statistically significant (e.g. for $n = 100$, in paired t-test p=0.00017). The difference between Docmap and LSI, although larger on the average, was not statistically significant due to considerably larger variation.

In effect, the model vectors representing clusters of similar documents obtained by the WEBSOM method are smoothed (i.e., generalized) representations of possible (sub-)topics of interest. The two-stage search thus first selects a small number of relevant (sub-)topics, then among those, picks the most relevant documents. Thus, the pre-selection phase serves to discard documents that have only accidental resemblance to the query, since they belong to a sub-topic that does not as a whole seem relevant enough for the query.

Conclusions

We have demonstrated that it is possible to scale up the SOMs in order to tackle very large-scale problems. Additionally, it has transpired that the encoding of documents for their statistical identification can be performed much more effectively than believed a few years ago [2]. In particular, the various random-projection methods are as accurate in practice as the ideal theoretical vector space method, but much faster to compute than the eigenvalue methods (e.g., LSI) that have been used extensively to solve the problem of large dimensionality.

Finally it ought to be emphasized that the order that ensues in the WEBSOM may not represent any taxonomy of the articles and does not serve as a basis for any automatic indexing of the documents. The strength of the large map displays is in "finding" rather than "searching for" relevant information. Nevertheless, experiments on a small reference collection indicate that the obtained clusters may serve as meaningful sub-topics that can be used to improve accuracy also in the search task.

References

[1] Salton, G., and McGill, MJ. *Introduction to modern information retrieval*. McGraw-Hill, New York, 1983.

[2] Deerwester, S., Dumais, S., Furnas, G., Landauer, K. Indexing by latent semantic analysis. *J Am Soc Inform Sci*, 1990; 41:391–407.

[3] Kaski, S. Dimensionality reduction by random mapping. *In Proc of IJCNN'98, Int Joint Conf on Neural Networks.* IEEE Press, Piscataway, NJ, 1998, pp. 413–418.

[4] Kohonen, T. New developments of learning vector quantization and the self-organizing map. *In SYNAPSE´92, Symposium on Neural Networks; Alliances and Perspectives in Senri 1992*, Osaka, Japan, June 24-26, 1992.

[5] Kohonen, T., Kaski, S., Lagus, K., Salojärvi, J., Honkela, J., Paatero, V., and Saarela, A. Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks*, vol. 11, number 3, pp. 574–585. May 2000.

[6] Lagus, K. Text Mining with the WEBSOM. *Acta Polytechnica Scandinavica, Mathematics and Computing Series No. 110*, 54 pp. December 2000. D.Sc(Tech) Thesis, Helsinki University of Technology, Finland.

[7] Lagus, K. Text retrieval using self-organized document maps. *Neural Processing Letters*. Accepted for publication. 2002.

[8] Lagus, K., and Kaski, S. Keyword selection method for characterizing text document maps. In *Proc. of ICANN99, Ninth Int. Conf. on Artificial Neural Networks* vol. 1, pp. 371–376. IEE, London, 1999.

[9] Voorhees, E. M., and Harman, D. K. Appendix: Evaluation techniques and measures. In *Proceedings of The Eighth Text REtrieval Conference (TREC 8)*. NIST, 2000.