In Proc. 19th Int. Conf. on Computational Linguistics (COLING), 2002.

Using Syntactic Analysis to Increase Efficiency in Visualizing Text Collections

James Henderson and **Paola Merlo** and **Ivan Petroff** and **Gerold Schneider**
University of Geneva, Geneva, Switzerland

Abstract

Self-Organizing Maps (SOMs) are a good method to cluster and visualize large collections of documents, but they are computationally expensive. In this paper, we investigate linguistically motivated reductions on the usual bag-of-words representation, to improve efficiency. We find that reducing the document representation to heads of verb and noun phrases reduces the heavy computational cost without degrading the quality of the map, especially in combination with term reduction techniques. More severe reductions which focus on subject and object nominal phrases are not advantageous.

1 Introduction

The recent considerable growth in the amount of easily available on-line text has attracted attention to the problem of obtaining readily usable information out of a very large unstructured collection of text documents. One step to a solution of this problem is to organize the documents into meaningful groups according to their content and to visualize the collection, providing an overview of the range of documents and of their relationships, so that they can be browsed more easily (Kohonen et al., 2000; Rauber and Merkl, 1999). Self-Organizing Maps (SOMs) (Kohonen, 1984) are an unsupervised method for generating a 2-dimensional visual map of a document collection. SOMs produce clusters of documents, which are positioned on the map such that similar clusters are next to each other. These clusters can then be labeled with their most important topics, giving an overview of the major topics covered in the document collection, and of their similarity to each other. Clustering the documents facilitates retrieval of the information that the user is looking for, while the spatial organization of the map supports the discovery of unlooked for, but related, pieces of information, like in an ordinary library, where books on similar topics are usually grouped in the same section (Rauber and Merkl, 1999).

The main advantages of using this method to create a visualization of the documents is that, compared to other methods, it is computationally feasible and it produces qualitatively better maps. Moreover, SOMs can integrate new incoming documents without recomputing the complete map every time (Kohonen et al., 2000). The main disadvantage in using this method is that, although feasible, it is computationally intensive. SOMs are a globally optimizing algorithm, iteratively searching to optimize a merit function defined over the entire map. These methods, akin to k-means clustering, achieve good performance, but can take a long time to train. One of the main factors affecting the efficiency of the algorithm is the size of the document representation.

In this paper, we investigate variations on the bag-of-words document representation usually employed in SOMs, in order to help reducing its size with minimal loss of information. In particular, we explore linguistically-motivated ways of selecting the most salient words in a document on the basis of a word's syntactic position in its sentence. Compared to a bag-of-words baseline model, we find that selecting the heads of noun and verb phrases yields a 19% reduction in computation time, with no loss in the quality of the map. More drastic linguistically-based reduction techniques, aimed at selecting only those terms that carry topic information, such as subjects, halve computation time, but reduce the quality of the maps significantly. We also compare these methods for selecting the important word tokens in individual documents with a simple method for selecting important word types across documents, namely selecting only the words with high document frequency. This frequency-based model produced a map which was of almost as good quality as that of the head-based model. Combining head-based word token selection with frequency-based word type selection produced a map with only slightly worse quality, but a 48% reduction in computation time over the bag-of-words baseline.

2 Computational Efficiency

The efficiency of the SOM training algorithm depends on the size of the document representations used. Documents are represented by a vector of values, as, for example, in the bag-of-words document representation, where each element in the vector is associated with a specific word. The document collection can be represented as a matrix, with each document's vector forming a row of the matrix. SOM training takes this matrix and iteratively searches for an optimal 2-dimensional map of clusters, a process which typically requires hundreds of iterations.

The time complexity of each iteration is:

$$O(\ |C|^2 \times |T| \ + \ |C| \times |V|\ )$$

where $|C|$ is the number of cluster positions in the map, $|T|$ is the number of different words used in the representation of all the documents, and $|V|$ is the number of values in the representation of all the documents. $|C|$ can be kept fairly small (we use 32 map positions), but $|T|$ can be large (11,606 for our baseline model). $|V|$ depends on how we represent the set of documents. If we represent the set of documents by a full document-by-word matrix, then the number of values in this matrix $|V'|$ would be the number of words $|T|$ times the number of documents. This method would make $|V'|$ huge (131,878,978 for our baseline model), and the computation of the SOM algorithm would be intractable. Fortunately, the document representation matrix is very sparse (99.7% of elements are zeros in our baseline model). By using a sparse matrix encoding of the document representation matrix, we can reduce $|V|$ to just the total number of non-zero values in the matrix (525,074 in our baseline model), thereby making the computation of the SOM algorithm tractable.

Even when the sparseness of the document-by-word matrix is exploited, training times for SOMs can be long, days or even weeks. In this paper we investigate ways of speeding up the training of SOMs by reducing the number of non-zero values $|V|$ in the document representations. A word's value in a document's representation becomes non-zero when an instance of that word is found in the document and counted. We investigate ways to choose which instances of words can be ignored and not counted, thereby reducing the number of non-zero values in the matrix. The difficulty with this approach is that ignoring words also potentially reduces the amount of information represented about the document, and thus could decrease the quality of the visualization produced by the SOM algorithm. We address this trade-off between efficiency and quality by using a syntactic analysis of the text to select which instances of words in the document are important for the document's representation.

Previous work on improving the speed of SOM training has focused on reducing the number of different words used in the representation of documents $|T|$. One approach is to apply Singular Value Decomposition to the document matrix, and only make use of the most important dimensions extracted. But this approach loses the sparseness of the document matrix, and the trade-off of the number of dimensions for sparseness is not advantageous (Kohonen et al., 2000). Two previous approaches which do not lose the matrix sparseness are to cluster words and use the clusters as the new smaller set of terms (Ritter and Kohonen, 1989), and to project the document matrix into a new smaller set of terms which are a random projection of the old set of words, but which maintain sparseness (Kohonen et al., 2000). The latter method in particular has had some success for very large document sets. Both these methods could be applied after those discussed in this paper to further improve efficiency.

Our approach of reducing the number of non-zero values also has the effect of reducing the number of words $|T|$. Some words are never selected for inclusion in any document's representation, and thus can be removed from the representation completely. However, this approach is still different from term selection methods, where a given word *type* is selected based on its distribution across the documents in which it appears. Our method selects word *tokens* based on the context in an individual document.

3 Identifying Important Words

The motivation behind using NLP techniques, in general, to select informative words in a text is that the importance of a word *token* depends on its type and on the specific linguistic context in which it appears. Syntactic analysis is a computationally efficient first step to identify which words bear contentful information in the document, under the assumption that there is a regular mapping between the content of a text and its syntactic structure. Because of current NLP technology's limitations, we choose to use those parts of a syntactic analysis that can be performed accurately on a large scale. Therefore, we tag the words, extract heads of phrases, as the identification of phrases is accurate, and identify subjects and objects, a task that can take advantage of the rather fixed word order of English, especially for subjects.

We experiment with four models. For comparison we also report results for a baseline model, which is a tagged lemmatized bag-of-word model. Model 1 reduces the document representation because only nouns and verbs that are heads of phrases are kept, while functional words and modifiers and words that are not heads are discarded. We expect this bag-of-heads representation to still capture the denotational and predicative content of the document, but to be considerably smaller in size, because the descriptive and qualitative aspects of it are discarded. Models 2, 3, and 4 explore increasingly drastic reductions to the set of words used in the representation. These models are motivated by a salience hierarchy based on grammatical function (Keenan and Comrie, 1977), which has been used successfully before for text summarization (Boguraev and Kennedy, 1997). According to this hierarchy, subjects are more salient than objects, which are more salient than other noun phrases. Model 2 differs from Model 1 in that nouns which are not in either subject or object position are not included in the document representation. Model 3 reduces the document representation further by also removing verbs, thereby representing a document as a bag of noun heads in subject or object position. Model 4 applies the most severe reduction and represents documents as a bag of noun heads in subject position.

From a linguistic point of view, our work is similar to (Hatzivassiloglou et al., 2000), who explore the use of noun phrase heads and proper names to enrich the feature set input to a hierarchical clustering algorithm. They *add* these features to the bag-of-words document representation, with the expectation that it will facilitate the algorithm in finding relevant terms. Their results are mixed: they find that the additional features improve overall clustering performance if used in combination with the initial words, but they also find an unexpected negative correlation between the head nouns and the topic clusters, which requires further investigation. Most other uses of NLP techniques in document processing and in particular in information retrieval, have aimed at enriching the document representation or the set of indexing terms, with mixed results (Lewis and Sparck-Jones, 1996; Strzalkowski, 1999). Differently from these pieces of work, we pursue here an application more aimed at visualizing documents than at ranking them, where NLP is used to *reduce* the complexity of the representation of the document, and to focus only on the important words for efficiency reasons. Therefore, we do not enrich the baseline representation, but we substitute it with more compressed models.

4 Methodology

Our data collection consists of the training portion of the Lewis Split of the Reuters-21578 database, for a total of 13,625 documents, varying from one sentence to several pages in length. The syntactic analysis was performed using the IPS system, a large-scale grammar-based parser that outputs very richly annotated structures (Wehrli, 1997). We use only a small portion of this annotation in our document representation models.

4.1 Implementing the Models

The **baseline** model is a tagged lemmatized bag-of-words representation. It utilizes the part of speech tags output by the parser to disambiguate word senses that can be detected by POS tag alone. A small hand evaluation over 882 words has revealed a tagging error of 6.3%.

Model 1 is based on the full syntactic analysis of the text produced by the IPS system. Specifically, we extract the head of all NPs and VPs in the document. Proper nouns are treated as multi-head phrases: we keep all their component words, as they all equally contribute to the meaning of the phrase. IPS hypothesizes proper nouns based on lexical information and on orthography and filters out many incorrect hypotheses while parsing. A small hand evaluation on 721 heads (4 articles) yields 94.3% precision and 98.1% recall for this step and 94% precision and 87.8% recall for recognition of proper nouns, on a sample of 100 items.

Models 2, 3, and 4 are also based on a full parse. In a structure-based syntactic analysis, different grammatical functions are defined by structural positions. The subject is the nominal phrase attached directly under the main sentential node, while objects occur directly inside the verb phrase, as a sister to the verb. Since proper nouns have been found to be particularly decisive topic indicators (Strzalkowski et al., 1995) we have again decided to include them disregarding their grammatical function. A small hand evaluation on 101 reported subjects (12 articles) yields 51.4% precision and 62% recall. For 92 reported objects, it yields 47.8% precision and 53% recall.

4.2 Computing the Document Vectors

As is standard in Information Retrieval (Salton and Buckley, 1988), each document is represented by a vector of term frequencies, weighted with inverse document frequency to reflect the importance of each term (called a tfidf vector):

$$v(d,t) = \mathrm{tf}(d,t) \times \ln(|D|/\mathrm{df}(t))$$

where d is the document, t is the term (a tagged lemma in our case), $\mathrm{tf}(d,t)$ is the number of term instances in d which are t, $|D|$ is the number of documents, and $\mathrm{df}(t)$ is the number of documents which contain t. Terms from a specific list of "stop words" (such as function words) are not included in the representation. Also, terms which occur in three or fewer documents are removed from the document representation, because these terms are too infrequent to have any impact on the results of the SOM algorithm, and removing them greatly reduces the total number of different terms $|T|$ (by 70% in the baseline model).

4.3 Training the Self Organizing Maps

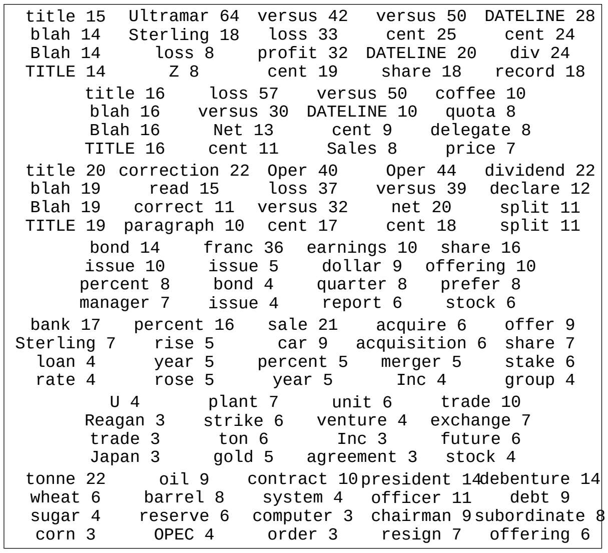
Given a set of document representation vectors, the SOM algorithm finds a partitioning of those documents into clusters and an assignment of these clusters to positions on a 2-dimensional grid. The range of documents in the collection can then be visualized by displaying the topic of each cluster on a 2-dimensional map, as illustrated in figure 1. The algorithm searches the space of clusterings and the space of position assignments simultaneously, trying to find a global optimum for two criteria. The first criterion is that clusters which are next to each other on the map (called "neighbors") have similar documents. This property means that the topics of clusters change continuously as one moves across the map, making it easier for a viewer to understand the range of documents in the collection than would be possible with an unstructured list of topics. The second criterion is that the documents within a given cluster are similar to each other. This property means that each cluster has a coherent topic.

More precisely, the SOM algorithm finds a "center" vector for each position on the given 2-dimensional grid. These center vectors specify the partitioning of the documents into clusters; a document vector is assigned to the cluster whose center vector is the closest. The similarity between neighboring clusters on the map is defined as the distance between their two center vectors, and the similarity between the documents in a cluster is defined as the average distance between a document vector and its center vector. Given an initial assignment of centers to map positions, the SOM algorithm iteratively adjusts the values of the center vectors in search of an assignment which optimizes both the criteria discussed above. [1]

4.4 Producing the Visualizations

The SOM's 2-dimensional grid of map positions lends itself naturally to a visual display, each cluster being assigned a position on the display according to its position in the grid. To summarize the topics of the documents in a cluster, we display a short list of the most important terms for characterizing that cluster, as illustrated in figure 1. The importance of a term is measured as the average value of the term across the document vectors in the cluster, minus the average value of the term across the document vectors in non-neighboring clusters. The first

[1] We used the "Batch-Map" (Kohonen et al., 2000) version of the SOM algorithm, with the cosine distance measure. The center vectors were initialized to points on the most important plane found by Singular Value Decomposition applied to the normalized document vectors, as recommended in (Kohonen et al., 2000). This method means that we start with the best linear projection onto a plane, and then allow non-linear optimization with the SOM algorithm. We chose the initial set of centers to reflect the distribution of documents, unlike in (Kohonen et al., 2000).



title 15 blah 14 Blah 14 TITLE 14	Ultramar 64 Sterling 18 loss 8 Z 8	versus 42 loss 33 profit 32 cent 19	versus 50 cent 25 DATELINE 20 share 18	DATELINE 28 cent 24 div 24 record 18
title 16 blah 16 Blah 16 TITLE 16	loss 57 versus 30 Net 13 cent 11	versus 50 DATELINE 10 cent 9 Sales 8	coffee 10 quota 8 delegate 8 price 7	
title 20 blah 19 Blah 19 TITLE 19	correction 22 read 15 correct 11 paragraph 10	Oper 40 loss 37 versus 32 cent 17	Oper 44 versus 39 net 20 cent 18	dividend 22 declare 12 split 11 split 11
bond 14 issue 10 percent 8 manager 7	franc 36 issue 5 bond 4 issue 4	earnings 10 dollar 9 quarter 8 report 6	share 16 offering 10 prefer 8 stock 6	
bank 17 Sterling 7 loan 4 rate 4	percent 16 rise 5 year 5 rose 5	sale 21 car 9 percent 5 year 5	acquire 6 acquisition 6 merger 5 Inc 4	offer 9 share 7 stake 6 group 4
U 4 Reagan 3 trade 3 Japan 3	plant 7 strike 6 ton 6 gold 5	unit 6 venture 4 Inc 3 agreement 3	trade 10 exchange 7 future 6 stock 4	
tonne 22 wheat 6 sugar 4 corn 3	oil 9 barrel 8 reserve 6 OPEC 4	contract 10 system 4 computer 3 order 3	president 14 officer 11 chairman 9 resign 7	debenture 14 debt 9 subordinate 8 offering 6

Figure 1: Labeled map for Model 1.

component of this difference reflects the importance of the term within the cluster, and the second component reflects the extent to which this term distinguishes the cluster from other clusters. The neighboring cluster are excluded from this second component because we want the display to reflect the similarities between neighboring clusters on the map. To reflect the relative importance of the terms, we also display the value of the importance measure.[2]

5 Experimental Evaluations

To measure the effects of the reduced representation models on the SOM algorithm, we trained several SOMs and evaluated both their training efficiency and the quality of the resulting maps. Based on previous experience with the number of iterations required, for each model we ran the algorithm described in section 4.3 for 200 iterations.

5.1 Efficiency Comparisons

To estimate the effects on computation time of the different models, we used a timing program to run the SOM implementation on each model for ten iterations. As shown in the left panel of table 1, all the models result in significant speed-ups over the baseline model, particularly considering the long computation times involved. These increases in speed are directly proportional to the reduction in document representation size. Specifically, the percent speedup lies between the reduction in the number of terms $|T|$ and the reduction in the number of non-zero values $|V|$ in the document representations, as expected according to the complexity analysis in section 2.

5.2 Quality Comparisons

Measuring the effect of our changes to the document representation on the quality of the maps produced by the SOM algorithm is a difficult task. The SOM algorithm is an unsupervised algorithm, so there is no gold-standard to compare the results against. Since we are primarily concerned with achieving a reduction in the document representation, without degrading the quality of the map, our assumption will be that the best map is obtained by the richest representation, that is our baseline model, and we will compare the other maps to this one. The map produced by Model 1 is shown in figure 1.

First, we observe the similarity of the 4 maps produced by the reduced models compared to the baseline map. We see that the quality of the Model 1 map is not degraded, as indicated by the fact that almost all clusters in Model 1 have a correspondence in the baseline map. Moreover, the labels suggest that they are fairly coherent clusters. On the contrary, the maps produced by Models 2 through 4 are not as similar to the baseline (with about a third of the clusters not having an obvious match in the baseline map). The coherence of their clusters is also slightly worse.

Second, we calculate several quantitative indices of the quality of the map, reported in the right panel of table 1. The first column (WCS) indicates the quality of the individual clusters. These figures measure the extent to which the map satisfies the second criterion discussed in section 4.3, minimizing the average distance between a document vector and its center vector. Because we are using cosine distance, the larger the number the better.[3] As can be seen, Model 1 does not decrease in quality compared to the baseline, while there is a progressive degrada-

[2] We display the importance measure multiplied by 100 and rounded to an integer. Only the lemmas, not the tags, of each term are displayed.

[3] We measured all the similarities in tables 1 and 2 in the baseline space, thereby ensuring that they measure properties of the clusters and not properties of the spaces.

	Timing and Complexity (% of baseline)			Measures of Quality			
	Sec/Iteration	Number Terms	Non-Zero Values	WCS	BNS	RTR	RTCS
baseline	59.338	11450	510586	0.342	0.305	72.6%	431
Model 1	48.032 (19.1%)	9413 (17.8%)	401276 (21.4%)	0.339	0.326	74.0%	498
Model 2	37.871 (36.2%)	7940 (30.7%)	295265 (42.2%)	0.327	0.401	68.4%	425
Model 3	27.757 (53.2%)	6403 (44.1%)	190644 (62.7%)	0.316	0.423	64.2%	429
Model 4	22.634 (61.9%)	5526 (51.7%)	139666 (72.6%)	0.308	0.384	60.0%	558

Table 1: Comparison of the models. (WCS: Within Cluster Similarity, BNS: Between Neighbor Similarity, RTR: Reuters Topic Recall, RTCS: Reuters Topic Cluster Size.)

tion from Models 2 to 4.

The second column of quality measures (BNS) reflects the quality of the positioning of clusters on the map. These figures measure the extent to which the map satisfies the first criterion discussed in section 4.3, minimizing the average distance between the two center vectors of neighboring clusters. Again, larger numbers are better. This measure of the quality of the topology of the maps shows no clear trend across the four models, but all the reduced representations do better than the baseline.

The average topic recall values (RTR) shown in the third column of quality measures compare our clustering to the original labels of topic in the Reuters collection. The Reuters corpus comes with a set of predefined topic labels. While it cannot be expected that an unsupervised clustering method would discover such predefined topics, these topics do give us an indication of which documents are considered similar by human judges. We assume that documents which are given the same topic label should be considered similar. The SOM should place similar documents close together in the map, preferably assigning them to the same cluster. As a measure of how well the SOM does this for the topic classes, for each topic we first found the cluster with the largest number of the topic's documents, and then compared this number to the number of the topic's documents in non-neighboring clusters, simply ignoring all the documents which are in neighboring clusters. Model 1 performs better than the baseline, while there is a progressive degradation from Models 2 to 4. Because this is a recall measure, it is possible to get 100% by putting all the documents in a single cluster. As a check that none of the models are maximizing performance in this way, we also show the average size of the chosen cluster for each topic. These figures mostly confirm the trend of the recall figures, but indicates that the improvement of Model 1 over the baseline may be the result of choosing larger clusters.

Taking these different quality measures together, we conclude that there is no loss in map quality between the baseline model and Model 1, but there is a progressive loss in quality when moving to Models 2 through 4. In addition, we note that the drop in quality from Model 2 to Model 3 suggests that verbs are important for text mining, contrary to the common belief for information retrieval.

6 Comparison with Term Selection Methods

Given the success of Model 1 at reducing the document representation without harming map quality by a linguistically-based selection of word tokens, we compare Model 1 to a methods for selecting word types based on frequency. This frequency-based model is the same as the baseline model except terms which occur in 42 or fewer documents are removed. This threshold was chosen because it produces a document representation with the same number of non-zero values as Model 1, as shown in table 2. The frequency-based model is faster than Model 1, due to its fewer terms. As can be seen in table 2, its map quality is equivalent to that of the baseline model, and it is also equivalent to Model 1, except for a slight reduction in the quality of the topology of the map (BNS).

These two methods for reducing the document representation size are very different, and yet they result in roughly equivalent performance of the SOM algorithm. It is thus natural to consider combining them. We derived a new model by taking Model 1 and removing

	Timing and Complexity (% of baseline)			Measures of Quality			
	Sec/Iteration	Number Terms	Non-Zero Values	WCS	BNS	RTR	RTCS
baseline	59.338	11450	510586	0.342	0.305	72.6%	431
Model 1	48.032 (19.1%)	9413 (17.8%)	401276 (21.4%)	0.339	0.326	74.0%	498
frequency	37.377 (37.0%)	2083 (81.8%)	401461 (21.4%)	0.340	0.304	73.5%	475
combined	30.642 (48.4%)	1772 (84.5%)	316145 (38.1%)	0.340	0.304	69.7%	431

Table 2: Comparison of Model 1, frequency-based term selection and a combination of the two models. (WCS: Within Cluster Similarity, BNS: Between Neighbor Similarity, RTR: Reuters Topic Recall, RTCS: Reuters Topic Cluster Size.)

all those terms which were not included in the frequency-based model. This resulted in a much smaller document representation, and a computation time which is almost half compared to those of the baseline model, as indicated in the last line of table 2. The quality of the map produced from this model is also equivalent to the baseline, except for some reduction in the correspondence between the clusters found and those defined by the Reuters topics. This indicates that the combination of term selection methods with linguistically-based word token selection methods is an interesting direction for future investigation.

7 Conclusions

These experiments show that we can achieve a significant increase in efficiency in visualizing text collections, without degradation of the maps, by representing documents with the heads of the more important parts of speech (Model 1). This confirms our initial intuition that denotational and predicative information is sufficient to characterize a document. On the other hand, the degradation observed in models that focus only on salient words (Models 2 to 4) indicates that the reductions in these models are too drastic. The comparison with a frequency-based model shows that the linguistically-based token reduction results in maps of equivalent quality to those produced by a drastic document frequency cut-off, and that a combination of these two methods yields promising initial results.

Acknowledgments

This research was supported by the Swiss NSF, grant 21-59416.99. Thanks to our colleagues, Abderrahim Labbi, Christian Pellegrini, and Eric Wehrli.

References

B. Boguraev and C. Kennedy. 1997. Salience-based content characterisation of text documents. In *Procs of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization*, pages 2–9, Madrid, Spain.

V. Hatzivassiloglou, L. Gravano, and A. Maganti. 2000. An investigation of linguistic features and clustering algorithms for topical document clustering. In *SIGIR 2000*, pages 224–231.

E. L. Keenan and B. Comrie. 1977. Noun phrase accessibility and universal grammar. *Linguistic Inquiry*, 8:62–100.

T. Kohonen, S. Kaski, K. Lagus, J. Salojarvi, J. Honkela, V. Paatero, and A. Saarela. 2000. Self organisation of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574–585.

T. Kohonen. 1984. *Self-Organization and Associative Memory.* Springer-Verlag, Berlin.

D. Lewis and K. Sparck-Jones. 1996. Natural language processing for information retrieval. *CACM*, 39(1):92–101.

A. Rauber and D. Merkl. 1999. The SOMLib digital library system. In *Procs. of the 3rd Europ. Conf. on Research and Advanced Technology for Digital Libraries (ECDL'99)*, Paris, France.

H. Ritter and T. Kohonen. 1989. Self-organizing semantic maps. *Biological Cybernetics*, 61(4):241–254.

G. Salton and C. Buckley. 1988. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523.

T. Strzalkowski, J. Perez Carballo, and M. Marinescu. 1995. Natural language information retrieval: Trec-3 report. In *NIST Special Publication 500-225*, pages 39–53, NIST, Gaithersburg, MD.

T. Strzalkowski, editor. 1999. *Natural Language Information Retrieval.* Kluwer Academic Publishers, Dordrecht.

E. Wehrli. 1997. *L'analyse syntaxique des langues naturelles.* Masson, Paris.