Using NLP to Efficiently Visualize Text Collections with SOMs

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

Abstract

Self-Organizing Maps (SOMs) are a good method to cluster and visualize large collections of text documents, but they are computationally expensive. In this paper, we investigate ways to use natural language parsing of the texts to remove unimportant terms from the usual bag-of-words representation, to improve efficiency. We find that reducing the document representation to just the heads of noun and verb phrases does indeed reduce the heavy computational cost without degrading the quality of the map, while more severe reductions which focus on subject and object noun phrases degrade map quality.

1. Introduction

Clustering large collections of documents and visualizing them appropriately can play a crucial role in effective and informative information retrieval [4, 7]. Self-Organizing Maps (SOMs) [3] are an unsupervised method for clustering and generating a 2-dimensional visual map of a document collection. Similar clusters are positioned next to each other, so that, when labeled with their most important topics, they give an overview of the major topics covered in the collection, and of their similarity to each other. Although SOMs are among the most effective methods to create and display maps, they can be computationally very intensive. One of the main factors affecting the efficiency of the algorithm is the size of the document representation. In this paper, we investigate linguistically-motivated variations on the bag-of-words representation usually employed in SOMs, to reduce the size of the document representations with minimal loss of information. In particular, we explore the effect on the clustering and map quality of selecting the most salient words in the documents on the basis of the syntactic structure of the sentences in the text and of the category of the word. We find that selecting the noun and verbs heads of the main syntactic phrases yields an 18% reduction in computation time, with no net loss in the quality of the maps. More drastic term reduction techniques, aimed at selecting only those noun phrases that carry topic information, such as subjects, halve computation time, but reduce the quality of the maps significantly.

2 Computational Efficiency

In SOMs, each document is represented as a vector of weighted word counts, and the entire collection as a matrix, whose rows are the individual document's vectors. 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). Because we use a sparse matrix encoding of the document representation matrix, $|V|$ is the total number of non-zero values in the matrix (525,074 in our baseline model). Although very large, this number is a huge decrease from what it would be if we also explicitly represented the zero values, due to the fact that the document representation matrix is very sparse (99.7% of elements are zeros in our baseline model).

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.

3 Identifying Important Words

The motivation behind using NLP techniques, in general, and parsing in particular, to select informative words in a text is that the importance of a word *token* depends on the specific linguistic *context* in which it appears. We experiment with three models, that are compared to a baseline which is a tagged lemmatized bag-of-word model.

Model 1 represents a document using only those nouns and verbs that are heads of phrases. Thus, it discards all adjectives and adverbs, and also those nouns that are not heads, such as modifiers in noun compounds. We expect this 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 and 3 represent a document using only its most salient nominal expressions, subjects and objects, which are good indicators of topic. According to the salience hierarchy [2], subjects are more salient than objects, which are more salient than other noun phrases. Thus, Model 2 is an intermediate model that represents the documents as a bag of heads in object and subject position. Model 3 applies the most severe reduction and represents documents as a bag of heads in subject positions.

4 Methodology

Materials for our experiments are the training portion of the Lewis Split of the Reuters-21578 database (13,625 documents). The syntactic analysis was performed using a large-scale grammar-based parser [11].

Implementing the Models The baseline model is a tagged lemmatized bag-of-words representation. A 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. Specifically, we extract the head of all NPs and VPs in the document.[1] Since proper nouns are considered multi-head phrases, we keep all their component words. A 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 and 3 determine subject and objects by looking at specific structural positions in the fully parsed output. Since proper nouns have been found to be particularly decisive topic indicators we have again decided to include them disregarding their grammatical function. A 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.

Computing the Document Vectors As is standard in Information Retrieval [9], 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 representation). Terms from a specific list of "stop words" (such as function words) are not included in the representation, as well as terms which occur in three or fewer documents. 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).

Training and Visualizing SOMs 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 below in figures 1 through 3. 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.[2]

To summarize the topics of the documents in a cluster, we display a short list of the most important terms for characterizing that cluster. The importance of a term is measured as the average value of the term across the document

[1] Prepositions are also kept, but most of them are later eliminated by the use of a stop word list.

[2] We used the "Batch-Map" [4] version of the SOM algorithm, with the cosine distance measure. The center vectors were initialized to randomly selected document vectors projected onto the most important plane found by Singular Value Decomposition (SVD) applied to the normalized document vectors. This method means that we start with the best linear projection onto a plane, and then allow non-linear optimization with the SOM algorithm.

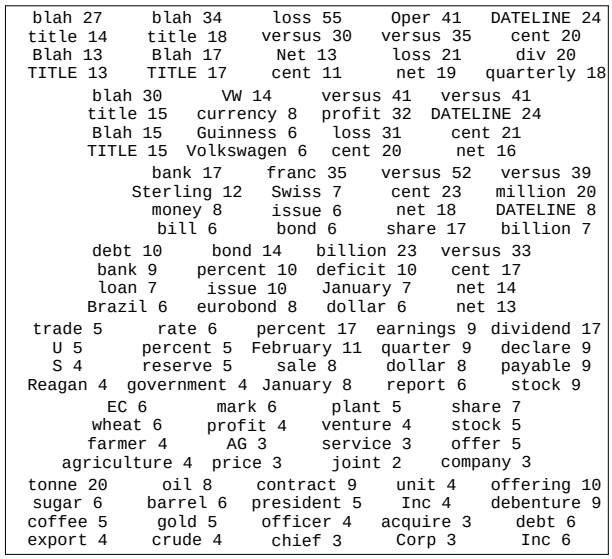

Figure 1. Labeled map for the baseline model.

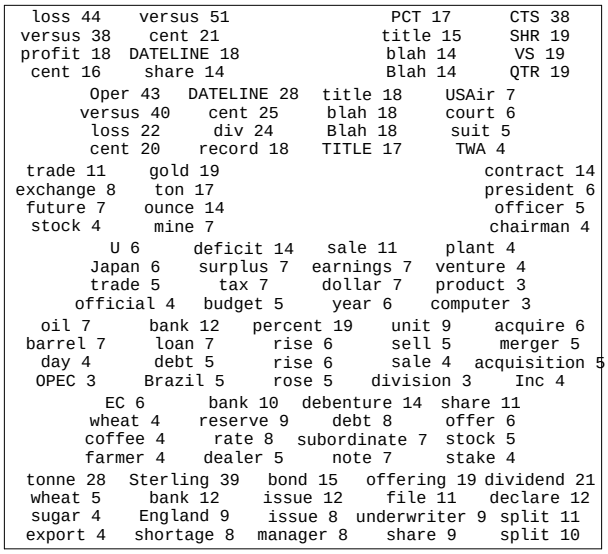

Figure 2. Labeled map for Model 1.

vectors in the cluster, minus the average value of the term across the document vectors in non-neighboring clusters. The first 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 unsimilar 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. The motivation behind this labeling method is similar to that of [5], but the formula is simpler. To reflect the relative importance of the terms, we also display the value of the importance measure (times 100). Only the lemmas, not the tags, of each term are displayed. Some of the labels found with this method actually reflect document formats rather than document topics. For example, the presence of the labels "blah" and "title" are due to the presence of documents which consist of only a title plus the text "blah blah blah".

5 Experimental Evaluations

To measure the effects of the reduced representation models on the SOM algorithm, we trained several SOMs for each model and evaluated both their training efficiency and the quality of the resulting maps. For each model, we ran the SOM algorithm described in section 4 three times, for 50, 100, and 200 iterations, respectively. Because the number of iterations had little impact on the models' relative performance, we report only the results for 200 iterations.

5.1 Efficiency Comparisons

As shown in table 1, all three models result in significant speedups over the baseline model. These increases in speed are important considering the long computation times involved, and as expected they are directly proportional to the reduction in document representation size.[3]

5.2 Quality Comparisons

Evaluating the quality of the maps is difficult, as 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. Some of the maps produced by the models are shown in figures 1 through 3.

First, we observe the similarity of the 3 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. Actually, we find an improvement over the baseline, as fewer clusters are labeled with irrelevant words. On the contrary, the maps produced by Models 2 and 3 are not as similar to the baseline, and the coherence of the clusters is not as good.

[3] Specifically, the percent speedup lies between the reduction in the number of terms $|T|$ and the reduction in the number of nonzero values $|V|$ in the document representations, as expected according to the complexity analysis in section 2.

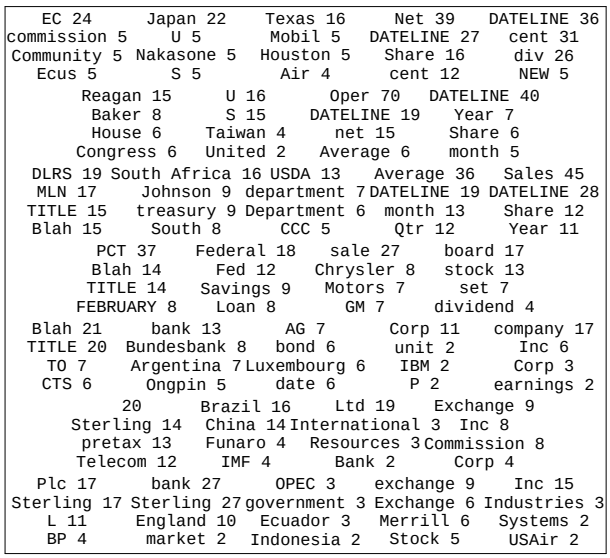

Figure 3. Labeled map for Model 3.

	Sec/It	Doc Terms	DNV
Baseline	427	11606	525,074
Model 1	350(18.1)	9546(17.1)	412,568(21.4)
Model 2	216(49.5)	6937(40.2)	192,598(63.3)
Model 3	178(58.2)	5826(49.8)	139,816(73.4)

Table 1. Timing and complexity of the models: second per iteration, document terms, and document non-zero values. (Percentage reduction from the baseline model.)

Second, we calculate several quantitative indices of the quality of the map, reported in Table 2. The first data column of the table (BNS) measures the quality of the positioning of clusters on the map. These figures measure the extent to which the map satisfies the first criterion used in training (see section 4), minimizing the average distance between the two center vectors of neighboring clusters. Because we are using cosine distance, the larger the number the better. This measure indicates that there is a progressive degradation in the quality of the topology of the map as the representations are reduced from the baseline. However, the degradation for Model 1 is not enough to imply a net loss in quality when balanced against the other measures.

The second data column of the table (WCS) indicates the quality of the individual clusters. These figures measure the extent to which the map satisfies the second training criterion (see section 4), minimizing the average distance between a document vector and its center vector. Again, larger numbers are better. As can be seen, Model 1 does not decrease in quality compared to the baseline, while Models 2 and 3 do.

	BNS	WCS	RTR	RLF
Baseline	0.394	0.335	76.1	8.19
Model 1	0.311	0.336	73.4	9.35
Model 2	0.240	0.288	49.9	3.66
Model 3	0.209	0.294	49.8	3.71

Table 2. Quantitative measures of performance: Between Neighbor Similarity (BNS), Within Cluster Similarity (WCS), Reuters' Topic Recall (RTR), Reuters' Topic $F_{\beta=1}$ (RTF).

The third and fourth columns of table 2 (RTR, RTF) 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 topics' documents, we first found the best cluster for each topic, and then compared the number of the topic's documents in this cluster to the number in non-neighboring clusters, simply ignoring all the topic's documents which are in neighboring clusters. This is a measure of topic recall (RTR). Model 1 performs almost as well as the baseline, while Models 2 and 3 show a degradation. However, if we combine recall with precision (RTF), we actually find that Model 1 improves slightly over the baseline.[4]

Combining the efficiency and the qualitative evaluation, we conclude the Model 1 is a successful attempt to reduce the size of the document representation without loss in quality of the output map, while the representations of Models 2 and 3, although greatly efficient, degrade quality too much.

6 Related Work

The main contribution of this work lies in using NLP techniques to achieve document compression, to support efficient use of complex visualization techniques.

[4]These $F_{\beta=1}$ values are dominated by the precision scores, which are very low because the clusters are much larger than the classes defined by topics. The relative improvement of Model 1 indicates that the clusters chosen for the topics are smaller than those in the baseline model, probably because there are fewer irrelevant clusters in Model 1. This difference would also account for the slight relative degradation of Model 1 on the RTR score.

Most other uses of NLP techniques in document processing, have aimed at enriching the document representation or the set of indexing terms, for document clustering [1] or information retrieval, [6, 10], with mixed results. 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.

Previous non-linguistic work on improving the speed of SOM training has focused on reducing the number of different words used in the representation of documents. One approach is to apply Singular Value Decomposition to the document matrix, and only make use of the most important dimensions extracted. Unlike ours, this approach loses the sparseness of the document matrix, and the trade-off of the number of dimensions for sparseness is not advantageous [4]. 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 [8], 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 [4]. 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 method also differs from term selection methods, where a given word *type* is selected based on the statistical distribution across the documents in which it appears. We select word *tokens* based on the context in an individual document. This approach, however, also has the effect of removing those word types that are never selected for inclusion in any document's representation.

7 Conclusions and Outlook

These experiments show that we can achieve a significant increase in efficiency, 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 2 and 3 indicates that the reduction in these models is too drastic. However, we should note that there is also the possibility that the poor performance of Models 2 and 3 are due to the error rate of the NLP tools used to detect subjects and objects, which is higher than that for detecting heads. Answering this question awaits the development of better NLP tools for performing this annotation.

The results of all the three models taken together indicate that there is an area of representations that could be profitably explored. Specifically, two main factors differentiate Model 1 from Model 2; in Model 2, verbs have been eliminated and common nouns have been reduced by half (from 214,017 to 107,742). In Model 3, the common nouns have been nearly reduced by half again, yet the performance of Models 2 and 3 are rather similar. This seems to indicate that it is the removal of verbs which has had the greatest impact on the performance of Models 2 and 3. Models that retain verbs while using subject and object syntactic roles to select nouns may allow the representation of the document to be further reduced, with a consequent improvement in the efficiency of the SOM algorithm, without degradation in quality of the final result.

Acknowledgments

This research was supported by the Swiss National Science Foundation, grant 21-59416.99. We would like to thank Christian Pellegrini and Abderrahim Labbi for their help in developing the ideas explored in this paper and Eric Wehrli for the use of the FIPS tagger and parser.

References

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

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

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

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

[5] K. Lagus and S. Kaski. Keyword selection methods for characterizing text document maps. In *Proc. 9th Int. Conf. on Artificial Neural Networks*, pages 371–376, London, 1999.

[6] D. D. Lewis and K. Sparck-Jones. Natural language processing for information retrieval. *Communications of the ACM*, 39(1):92–101, 1996.

[7] A. Rauber and D. Merkl. The SOMLib digital library system. In *Proc. of the 3rd Europ. Conf. on Research and Advanced Technology for Digital Libraries*, Paris, 1999.

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

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

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

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