Evaluating Keyword Selection Methods for WEBSOM Text Archives

Arnulfo P. Azcarraga, *Member*, *IEEE*,
Teddy N. Yap Jr., Jonathan Tan, and
Tat Seng Chua, *Member*,
IEEE Computer Society

Abstract—The WEBSOM methodology, proven effective for building very large text archives, includes a method that extracts labels for each document cluster assigned to nodes in the map. However, the WEBSOM method needs to retrieve all the words of all the documents associated to each node. Since maps may have more than 100,000 nodes and since the archive may contain up to seven million documents, the WEBSOM methodology needs a faster alternative method for keyword selection. Presented here is such an alternative method that is able to quickly deduce meaningful labels per node in the map. It does this just by analyzing the relative weight distribution of the SOM weight vectors and by taking advantage of some characteristics of the random projection method used in dimensionality reduction. The effectiveness of this technique is demonstrated on news document collections.

Index Terms—Keyword extraction, text archives, WEBSOM, random projection.

1 Very Large Text Archives

TEXT archives now run in the order of millions of documents. With such sizes of text archives being processed, words that appear at least once in the text corpus, even after removal of very common (stop) words, run to more than 100,000! One important methodology that is effective in archiving up to seven million text documents is the WEBSOM [1], [2], [3], [4], [5], [6], [7], which uses a "Self-Organizing Map" (SOM) at the core of its archiving technique. A number of other SOM-based text archiving techniques have been described in the literature [8], [9], [10], [11]. They differ mainly in the preprocessing and postprocessing stages of the archiving process.

WEBSOMs need automatic procedures for extracting keywords of archived documents which are useful for browsing purposes. Extracting keywords for maps based on the WEBSOM methodology, however, is not straightforward because of a *random projection method* that is employed to compress the large but sparse input term frequency vectors. The WEBSOM methodology does include an automatic keyword extraction procedure [6], which we refer to as the "Lagus method." The procedure, however, is rather slow. It computes the relative frequencies of all the words of all the documents associated to each node and then compares these to the relative frequencies of words of the other nodes in the map. Since maps may have more than 100,000 nodes and the archive may contain up to seven million documents, the existing method is not practical. Another keyword extraction method has been reported [8], but this was not done on input vectors that have gone through random projection.

- *A.P. Azcarraga is with the School of Information Technology and Computing, De La Salle University-Canlubang, Laguna Blvd. Binan, Laguna, Philippines. E-mail: azcarragaa@dlsu.edu.ph*
- *T.N. Yap Jr., J. Tan, and T.S. Chua are with the PRIS Group, School of Computing, National University of Singapore, Lower Kent Ridge, Singapore 117543. E-mail dcstny@nus.edu.sg, dcsjot@nus.edu.sg, and chuats@comp.nus.edu.sg.*

Manuscript received 18 July 2001; revised 17 June 2002; accepted 28 Oct. 2002.
For information on obtaining reprints of this article, please send e-mail to: tkde@computer.org, and reference IEEECS Log Number 114547.

The alternative method we describe in this paper deduces the keywords by just analyzing the relative weight distribution of the SOM weight vectors and by taking advantage of some characteristics of the random projection method. Our method is several orders of magnitude faster than the current WEBSOM technique, and yet retrieves fairly the same keywords as the WEBSOM method.

Keyword selection is important because it allows SOMs to be used as a novel interface for navigating through a text archive [5], [7]. Keywords extracted as labels for clusters of documents give the user a clue as to what are contained in the documents. In addition, knowing the top keywords per unit allows the assigning of nonuniform weights to the different dimensions of centroids-based text classification algorithms [12], [13], [14]. Central to these techniques, of course, is an effective way of knowing which dimensions (i.e., keywords) should receive more weight. Likewise, when developing hierarchical SOMs [9], it is often useful to allocate different weight distributions to different layers of the tree. There again, it is important to know which are the central keywords. Finally, being able to explain why documents are grouped together is important in document clustering [15]. Again, the major keywords of each cluster is of critical value.

The rest of the paper is organized as follows: Section 2 describes the WEBSOM methodology and the random projection method. Section 3 describes and analyzes the process of deducing the most important keywords for each unit of the map. The effectiveness of the keyword deduction method is illustrated in Section 4, while Section 5 assesses the quality of keywords extracted by the proposed method.

2 WEBSOM Methodology for Text Archiving

The SOM training algorithm used by WEBSOM is among the most well-known neural network training algorithms with a large number of reported applications [16], [17], [18], [19]. In text archiving and retrieval, the most prominent use of SOMs has been the WEBSOM. In this text archiving methodology, a standard stopword removal and stemming preprocessing procedure are conducted on the text collection, after which a two-step dimensionality reduction process is performed prior to training the map. The individual nodes of the map are then labeled, and once labeled, the map is ready for browsing and searching.

The dimensionality reduction procedure includes the compression of the very high dimensional, but largely sparse term frequency vector using a *random projection* method that was reported in [1] by Kohonen and in [4], [7] by Kohonen et al. and studied by Kaski in [19]. Given a document vector $\boldsymbol{n}_i \in \Re^n$, where the elements of the vector are normalized term frequencies, and given a random $m \times n$ matrix $\mathbf{R}$ whose elements per column are normally distributed. One can compute the projection $\boldsymbol{x}_i \in \Re^m$ of the original document vector $\boldsymbol{n}_i$ on a much lower dimensional space (i.e., $m \ll n$) using $\boldsymbol{x}_i = \mathbf{R}\,\boldsymbol{n}_i$, where $\mathbf{R}$ is the *random projection matrix*. Each term is randomly mapped to $\boldsymbol{r}$ dimensions and each dimension, in turn, is associated with approximately $\boldsymbol{rn/m}$ terms, where $\boldsymbol{m}$ is the number of dimensions in the compressed input vector, and $\boldsymbol{n}$ is the original number of keywords prior to random projection. Despite compressing the number of dimensions to under 1 percent of the original number, most of the original information content necessary for effective text classification and archiving are preserved. This very elegant and efficient document encoding procedure is one feature of WEBSOM that truly stands out, despite the difficulty it presents as far as keyword extraction is concerned.

 Published by the IEEE Computer Society

1. For every dimension of the (compressed) reference vector, compute the mean weight μ and standard deviation σ among all the map units. All weight values that exceed $\mu+z\sigma$ are *significantly high* for the given dimension;
2. If a certain dimension d is significantly high, it is likely that only one of the many terms mapped to it has truly contributed significantly to the high weight. The rest of the terms are "*piggy-back*" terms which are associated with the significant keywords through random projection;
3. Since the random projection method randomly assigns each keyword to r different dimensions, then the common keywords will consistently contribute high weights to the r dimensions. Select those terms that have greater than r° significantly high dimensions. We use $r^\circ = 0.6 * r$;
4. To extract k keywords, take the top k terms from the list extracted from step 3 that has been sorted in decreasing order of their accumulated weights.

Fig. 1. Distribution of region codes based on the origin of the news documents. Also shown are the extracted keywords for some portions of the map.

3 Alternative Method for Extracting Meaningful Labels for Each Node

It must be underscored that document encoding by random projection is not exclusive to WEBSOM and so our method for extracting meaningful labels for document clusters is applicable to any centroids-based text archiving system that uses random-projection for document encoding. However, just so we can compare our method with an existing WEBSOM keyword extraction method, we limit discussions to WEBSOM archives in the examples and analysis that follow.

The distribution of the weights of every node relative to the weight distributions of other units in the map determines to which nodes the various text documents are associated during archiving. Everything else equal, those terms mapped to high weight values are more significant than those mapped to lower valued weights. Since we use random projection, however, where each weight component has numerous terms mapped to it, there is no straightforward way to determine which of the keywords truly contribute significantly to those high weights.

By doing a reverse mapping based on the random projection matrix, we are able to trace back the various combinations of terms that contribute to each dimension in the compressed input vector. From these combinations, we can deduce the set of truly significant keywords. We highlight the main steps in Fig. 1.

Since the number of dimensions is very small compared to the number of unique terms $\mathbf{N}$, our method spends most of its time on step 3, which has $\mathbf{O}(\mathrm{N})$ complexity. On the other hand, the Lagus method has to retrieve each word in every document of every node in the map. Assuming it is able to store all relevant statistics in one pass through the archive, the retrieval of documents is of order $\mathbf{O}(\mathbf{D})$, where $\mathbf{D}$ is the number of documents in the archive. For the computation of the G measure defined in Section 5 (1), the method requires roughly $\mathbf{O}(\mathbf{n}^2 \times \mathbf{N})$ operations, where $\mathbf{n}$ is the number of nodes in the map. In other words, the Lagus method is slower compared to our method by a factor of about $\mathbf{n}^2$. Furthermore, I/O operations are much slower to execute and the Lagus method has to execute at least $\mathbf{D}$ read operations. In comparison, our method does not have to make a single read operation.

4 Extracting Keywords for News Archives

Our keyword deduction technique was applied on a WEBSOM text archive of a 1999 CNN news collection. In this collection, each of the news documents is assigned a class $0, 1, \ldots, 4$ depending on where the news document originated from. The class codes are as follows: 0: Africa, 1: America, 2: Asia, 3: Europe, and 4: Middle East. This data set has 7,597 training documents with 33,795 unique keywords that appear in at least one document. Through feature selection, the number of dimensions is reduced from 33,795 to 13,630. The random projection process further reduced the number of dimensions to only 315.

The trained and labeled 16×16 WEBSOM is shown as a map of region codes 0 to 4 in Fig. 2. The numbers correspond to the regions from which at least 50 percent of the news documents of each node come from. Our keyword extraction technique yielded the keywords shown in the same figure. Comparing the extracted keywords with the region codes, the deduced keywords obviously give a much better picture of what the clusters of documents are pertaining to. For example, there is a group of documents that discuss the plight of Augusto Pinochet. The "Middle East section" at the lower left corner of the map are documents about Iraq and the Middle East conflict. Many of the European documents on the lower right corner of the map are about the war in Kosovo.

Since news stories that are reported in one continent, say Asia, may pertain to totally unrelated events such as the North-South Korea conflict and the Anwar Ibrahim arrest in Malaysia, the region code alone does not reveal much about the specific news stories. This is why automatically extracted keywords are better node labels than the region codes in terms of giving users an idea of what topics are discussed in the documents associated to each node. Extracted keywords are actually based on the relative frequencies of the words used in the documents, while manually assigned labels may be based on something else.

The keyword deduction technique was also applied to a WEBSOM text archive of a subset of the Reuters 21,578 news collection, a text collection that has been well-studied from the point of view of text classification. This text collection is significantly different from the CNN collection in that each document can be assigned to multiple categories. Also, the manually assigned category labels, such as "corn," "coffee," and "money-fx," are much more descriptive than the news origin labels of the CNN collection. Since some classes in the original Reuters collection have zero or just a handful of news stories per class, we have used only a subset of the original news collection. Our experiment on the trained SOM using both the entire news collection and the subset we used here was reported in [10]. Details about the extracted keywords for the Reuters subset can be found in [11]. As for the assessment of the quality of the keywords extracted based on both this news collection and the CNN collection, this is presented in the next section.

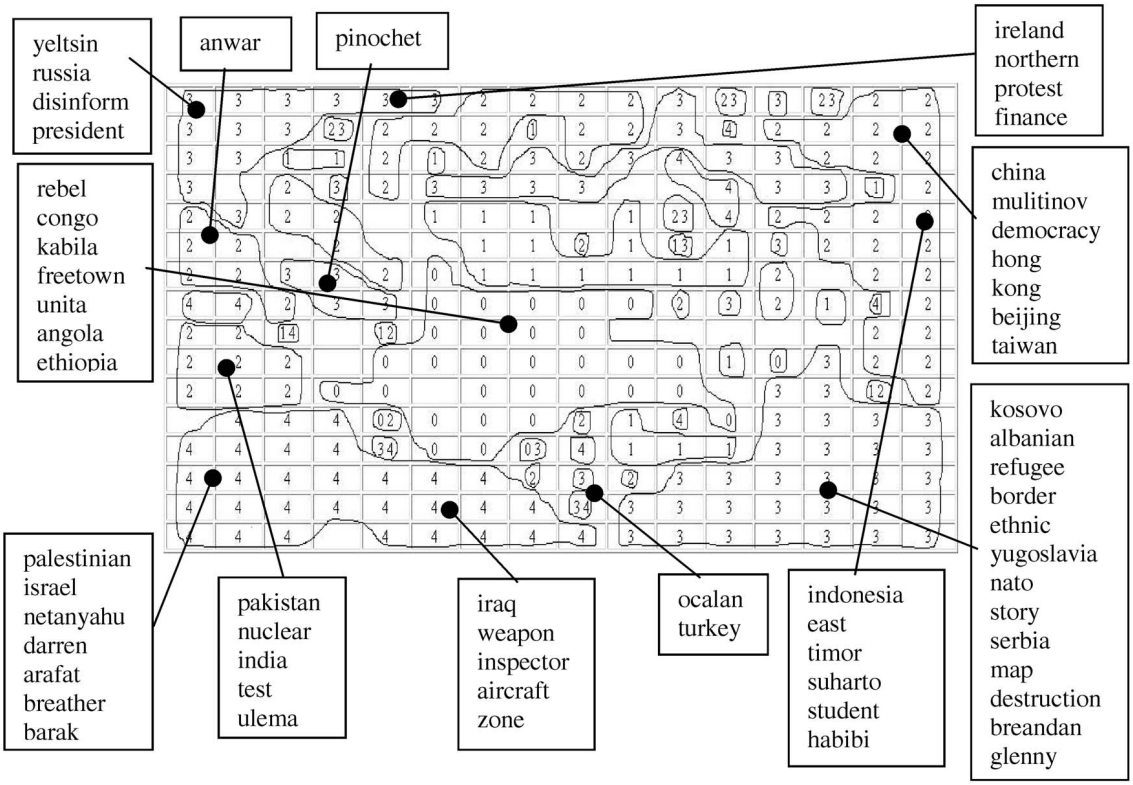

Fig. 2. Distribution of region codes based on the origin of the news documents. Also shown are the extracted keywords for some portions of the map.

5 Assessing the Keyword Extraction Method

The keyword extraction technique by Lagus and Kaski [6], against which our method is benchmarked in this section, directly computes the relative frequencies of occurrence of all words in all the documents assigned to a given node in the map. A goodness measure G, defined below, is used to rank the words as to how much they meaningfully represent a given node (in Lagus and Kaski [6], this is the G^2 measure):

$$G(w,j) = \left[\sum_{k\in R_j^0} F_k(w)\right] \frac{\sum\limits_{k\in R_j^0} F_k(w)}{\sum\limits_{k\in R_j^0} F_k(w) + \sum\limits_{k\in R_j^2} F_k(w)}, \qquad (1)$$

where R_j^0 is the region of units that form the same cluster as unit j and R_j^2 is the region of units much farther away from unit j, considered to be outside the cluster. The relative frequencies of each word w for a given unit k, denoted by $F_k(w)$, is the number of times the term w occurs in unit k normalized by the total number of occurrences of all words in all the documents assigned to unit k.

We used the Lagus method as the benchmark method[1] since the method is based on an exhaustive retrieval of all word lists of all documents associated to each and every node in the map. Table 1 presents the percentage of keywords extracted by our method that match the keywords extracted by the Lagus method for both the CNN and Reuters collections. Over all, for the two news collections we studied, we claim that our method extracts fairly the same keywords as the Lagus method. The fact that our method extracts more or less the same keywords is significant in that our method extracts keywords without digging out the individual word lists.

We still have not established, however, whether, in fact, the keywords extracted both by the Lagus method and our method are, in fact, meaningful. To circumvent the lack of an authoritative benchmark data set, we used the *titles* of news documents as the basis for assessing the quality of our extracted keywords. We take title words as the target keywords and that our keyword extraction method must be able to extract these title words. We argue that news reporters and editors know best as to which few words aptly describe each news story. Note that *precision* (i.e., proportion of extracted words that are also title words) is more important than *recall* (i.e., proportion of title words that are extracted as keywords) because some words in the title have functional roles, meant only to complete that thought evoked by the true keywords in the title.

There are 2,023 news articles in our Bveritas collection, with 21,090 unique stemmed words after stop-word removal and stemming. We then determined how much of the extracted keywords match with any of the title words in the documents that form the cluster for which the keywords have been extracted. Table 2 presents the average percentage match for the top 1, 3, and

TABLE 1
Percentage of the Top Three Keywords Per Node Extracted by Our Method that Match the Top 1, 3, or 8 Keywords Extracted for the Same Nodes Using the Lagus Method

	% match of extracted keywords							
	CNN Collection				Reuters Collection			
z-value	# keywords	1	3	8	# keywords	1	3	8
1.282	349	32	68	83	676	27	50	68
1.645	233	39	77	90	512	32	58	79
1.96	173	47	71	91	382	39	66	85
2.326	111	45	86	95	250	47	71	91
2.576	74	54	92	96	195	50	74	93
3.09	38	63	97	97	117	56	81	98
3.291	28	64	93	96	97	58	82	96

1. Work by Rauber and Merkl [8] assumes a one-to-one correspondence between the terms and the dimensions of the weight vectors, which is not true under random projection.

TABLE 2
Percent of Top Three Keywords Extracted Per Unit that Match the Top 1, 3, and 8 Keywords Extracted for the Same Units Using the Lagus Method, Based on the Bveritas Collection

		% match of keywords			% match with title words					
		*liGHt*SOM vs Lagus			*liGHt*SOM method			Lagus method		
z-value	# keywords	top 1	top 3	top 8	top 1	top 3	top 8	top 1	top 3	top 8
1.282	559	43	68	87	48	46	43	61	49	37
1.645	353	51	76	91	51	53	52	61	50	38
1.96	260	57	82	93	58	59	58	64	52	40
2.326	173	64	87	96	62	63	63	66	53	39
2.576	140	66	86	96	65	65	65	68	54	40
3.09	86	72	91	98	81	76	76	73	60	45
3.291	65	75	93	98	83	81	81	75	62	46

Also shown are the percentage matches with title words when considering the top 1, 3, and 8 keywords extracted by our liGHtSOM method and the Lagus method.

8 keywords. As a baseline figure, if words were selected at random, the estimated percentage match with title words is just 4 percent. The results confirm that the extracted keywords reflect the kinds of words that human experts would assign to news documents.

The best values that we found for the Lagus method, after trying out different values for the two radius parameters of the G^2 measure, are also shown in Table 2, along with the match rates between our method and the Lagus method. The percentage match rates are given on a per z-value basis, even for the Lagus method because only those nodes for which our method was able to extract keywords were included. This is a fairer comparison, otherwise, the Lagus method will register very low values (i.e., keywords with low G^2 scores would be included).

6 Conclusion

Keyword extraction methods for WEBSOM document archives, that have more than 100,000 nodes in the map and that may contain up to seven million documents, must avoid retrieving all the words of all the documents associated to each map unit. The method we describe in this paper deduces the keywords by just analyzing the relative weight distribution of the SOM weight vectors and by taking advantage of some characteristics of the random projection method used in dimensionality reduction prior to training.

We demonstrated the effectiveness of our technique by applying it on a WEBSOM archive of the CNN and Reuters text collections. The quality of our keyword extraction method is then methodically assessed by comparing the keywords extracted by our method with those extracted using the WEBSOM (Lagus) method. A high percentage of the keywords we extract match the top keywords extracted for the same nodes using the Lagus method. We conclude that our method is a fast alternative for WEBSOM-based archiving systems since, unlike the Lagus method, it does not need to dig out all the words of all the documents associated to every node in the map and yet extracts fairly the same keywords.

We likewise showed that a high percentage of our extracted keywords are, in fact, in the title of the news stories that we used in our experiments. Since the titles have been chosen by the newspaper editors and reporters, these are reasonable benchmark targets. We intend to use this keyword extraction method on a much larger online news integration archive that we are currently developing. Its extension to key-phrase extraction is also being explored.

References

[1] T. Kohonen, "Self-Organization of Very Large Document Collections: State of the Art," *Proc Int'l Conf. Artificial Neural Networks (ICANN '98)*, 1998.

[2] S. Kaski, T. Honkela, K. Lagus, and T. Kohonen, "WEBSOM—Self-Organizing Maps of Document Collections," *Neurocomputing*, vol. 21, pp. 101-117, 1998.

[3] S. Kaski et al., "Statistical Aspects of the WEBSOM System in Organizing Document Collections," *Computing Science and Statistics*, vol. 29, 1998.

[4] T. Kohonen et al., "Self-Organization of a Massive Document Collection," *Kohonen Maps*, Elsevier, 1999.

[5] K. Lagus et al., "WEBSOM for Textual Data Mining," *Artificial Intelligence Rev.*, vol. 13, pp. 345-364, 1999.

[6] K. Lagus and S. Kaski, "Keyword Selection Method for Characterizing Text Document Maps," *Proc. Ninth Int'l Conf. Artificial Neural Networks (ICANN '99)*, 1999.

[7] T. Kohonen et al., "Self-Organization of a Massive Document Collection," *IEEE Trans. Neural Networks*, vol. 11, no. 3, May 2000.

[8] A. Rauber and D. Merkl, "Automatic Labeling of Self-Organizing Maps: Making a Treasure Maps Reveal Its Secrets," *Proc. Fourth Pacific-Asia Conf. Knowledge Discovery and Data Mining (PAKDD '99)*, 1999.

[9] M. Dittenbach, D. Merkl, and A. Rauber, "Using Growing Hierarchical Self-Organizing Maps for Document Classification," *Proc. European Symp. Artificial Neural Networks (ESANN '00)*, 2000.

[10] A. Azcarraga and T. Yap Jr., "SOM-Based Methodology for Building Large Text Archives," *Proc. Seventh Int'l Conf. Database Systems for Advanced Applications (DASFAA '01)*, 2001.

[11] A.P. Azcarraga and T. Yap Jr., "Extracting Meaningful Labels for WEBSOM-Based Text Archives," *Proc. 10th ACM Int'l Conf. Information and Knowledge Management (CIKM '01)*, 2001.

[12] D.W. Aha, "Feature Weighting for Lazy Learning Algorithms," *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, eds., Norwell, Mass.: Kluwer, 1998.

[13] E.H. (Sam) Han and G. Karypis, "Centroid-Based Document Classification: Analysis and Experiment Results," *Proc Fourth European Conf. Principles of Knowledge Discovery and Data Mining (PKDD '00)*, 2000.

[14] S. Shankar and G. Karypis, "Weight Adjustment Schemes for a Centroid Based Classifier," *Text Mining Workshop, Proc Knowledge Discovery and Data Mining (KDD '00)*, 2000.

[15] D. Memmi and J.G. Meunier, "Using Competitive Networks for Text Mining," *Proc. Second Int'l ICSC Symp. Neural Computation (NC '00)*, 2000.

[16] T. Kohonen, "Self-Organized Formation of Topologically-Correct Feature Maps," *Biological Cybernetics*, vol. 43, no. 1, pp. 59-69, 1982.

[17] T. Kohonen, *Self-Organization and Associative Memory*, series in information sciences, second ed. Springer-Verlag, 1988.

[18] T. Kohonen, *Self-Organizing Maps*. Berlin, Springer-Verlag, 1995.

[19] S. Kaski, "Dimensionality Reduction by Random Mapping: Fast Similarity Computation for Clustering," *Proc. Int'l Joint Conf. Neural Networks (IJCNN '98)*, vol. 1, pp. 413-418, 1998.