Comparing Keyword Extraction Techniques for WEBSOM Text Archives

Arnulfo P. Azcarraga
PRIS Group, School of Computing
National University of Singapore
Lower Kent Ridge Road, Singapore 117543
(65) 874-6563
dcsapa@nus.edu.sg

Teddy N. Yap Jr.
College of Computer Studies
De La Salle University
2401 Taft Avenue, Manila, Philippines
(63-2) 524-0402
Teddy@CCS.dlsu.edu.ph

Abstract

The WEBSOM methodology for building very large text archives has a very slow method for extracting meaningful unit labels. This is because the method computes for the relative frequencies of all the words of all the documents associated to each unit and then compares these to the relative frequencies of all the words of all the other units of the map. Since maps may have more than 100,000 units and the archive may contain up to 7 million documents, the existing WEBSOM method is not practical. A fast alternative method is based on the distribution of weights in the weight vectors of the trained map, plus a simple manipulation of the random projection matrix used for input data compression. Comparisons made using a WEBSOM archive of the Reuters text collection reveal that a high percentage of keywords extracted using this method match the keywords extracted for the same map units using the original WEBSOM method.

1. Building Large WEBSOM Text Archives

"Self-Organizing Maps" (SOM), and most prominently the WEBSOM, have been shown to scale up to very large document collections [1-10]. However, being used mainly with data that are not pre-labeled, SOMs need automatic procedures for extracting keywords of archived documents if some information about the document clusters were to be given to the user. Knowing the top keywords per unit allows assigning non-uniform weights to the different dimensions of centroids-based classification algorithms [11, 12, 13]. Central to these techniques, of course, is an effective way of knowing which dimensions (keywords) should receive more weight. Likewise, in hierarchical SOMs [2, 3, 4, 5], it is useful to allocate different weight distributions to different layers of the tree. There again, it is important to know which are the central keywords per unit. Furthermore, being able to explain why certain documents are grouped together is important to studies on clustering of documents [14]. To do this, we should be able to isolate the major keywords that characterize each unit in the map. Finally, knowing the keywords associated with the units allows the user to view the label distribution and "guess" where the interesting documents are.

Extracting keywords is not straightforward because of a random projection method that is employed to compress the large but sparse input term frequency vectors. Some previous work has been done on keyword extraction for SOM-based archives [4, 5, 9]. In fact, the WEBSOM methodology does include an automatic keyword extraction procedure [9], but the procedure is very slow. It computes the relative frequencies of all the words of all the documents associated to each unit and then compares these to the relative frequencies of words of the other units of the map. Since current WEBSOM text archives have more than 100,000 units and may contain up to 7 million documents, the existing keyword extraction method is not practical.

This paper is organized as follows. Section 2 describes the process of deducing the most important keywords. The keyword deduction method is illustrated in section 3 using a WEBSOM-based archive of the well known Reuters text collection. Comparisons of our keyword selection technique with the original WEBSOM keyword selection method are presented in Section 4.

2. Extracting Meaningful Labels

The most critical aspect of SOM-based text archiving is the compression of the initial text dataset into a size that is manageable as far as SOM training, labeling, and archiving are concerned - this without losing too much of the original information content necessary for effective text classification and archiving.

First reported in Kohonen [8, 10], a *random projection method* can radically reduce the dimensionality of the document encodings. Given a document vector $\boldsymbol{n}_i \in \Re^n$, where the elements of the vector are normalized term frequencies after performing feature selection, and given a random m x 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 $\boldsymbol{x}_i = \mathbf{R}\, \boldsymbol{n}_i$.

Kohonen [8, 10] reports that the similarity relations between any pairs of projected vectors ($\boldsymbol{x}_i$, $\boldsymbol{x}_j$) are very good approximations of the original document vector pair ($\boldsymbol{n}_i$, $\boldsymbol{n}_j$) for as long as m is at least 100. Given r as the number of 1s per column in the random projection matrix, m as the number of dimensions in the compressed input vector, and n as the original number of keywords prior to random projection. Each term is randomly mapped to r dimensions. Each dimension, in turn, is associated with approximately rn/m terms. In our experiments with the Reuters collection, we used m=315, r=5 and n=2,920.

Before we describe our keyword extraction procedure, we need to be clear as to what a good keyword is. In general, we want these keywords to be meaningful labels for the individual units of the map so that a user who browses a WEBSOM-based text archive may have as good a picture as possible of the contents of the documents assigned to the individual units.

```
for d = 1 to m
   if w_qd ≥ μ_d + z.σ_d
         for j = 1 to n
              if RPM[d][j] = 1
                  add 1 to tallyFreq [j]
                  add w_qd to sumWeights [j]
              endif
         endfor
   endif
endfor

sortedTermIndex [] = sort (sumWeights[])

k=0; j=0
while (k < ExtractedKeywords and j < n)
   if tally_freq [sortedTermIndex [j]] ≥ r°
        output term [sortedTermIndex [j]]
       k=k+1
   endif
   j=j+1
endwhile
```

Procedure 1. Keyword extraction procedure

We adopt here the two principles used in Lagus [9] that intuitively define a meaningful label for a unit in a trained WEBSOM. A term w is a meaningful label for a document cluster C in a trained WEBSOM if 1) w is prominent in C compared to other words in C; and 2) w is prominent in C compared to the other occurrences of w in the whole collection.

The distribution of the weights of every map unit relative to the weight distributions of other units in the map determines where the various text documents are associated during archiving. Those terms mapped to high weight values are more significant than those mapped to lower valued weights. In other words, terms mapped to high weight values are the potential keywords for the documents associated to a given map unit. But since we used a random projection matrix, each weight component has numerous terms mapped to it. Thus, there is no straightforward way to determine which are the keywords that truly contribute significantly to the high weight value of a map unit.

If we study how the random projection method works, however, we would be 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 as follows:

1. For every dimension, compute the mean weight μ and standard deviation σ among all the map units. Weight values that exceed $\mu+\mathbf{z}\sigma$ are significantly high for the given dimension. For example, weights greater than $\mu+\mathbf{z}\sigma$, at $\mathbf{z}$=1.645, have 95% confidence of being significantly higher than the mean. Higher **z**-values imply higher confidence levels.
2. Every time a certain dimension d is found to be significantly high, it is likely that only one of the rn/m terms mapped to it has truly contributed significantly to the high weight of that unit. The rest of the terms are just "*piggy-back*" terms.
3. Since the random projection method randomly assigns each keyword to r different dimensions, then the truly significant keywords will consistently contribute high weights to the r dimensions. If we count how many of each term's randomly projected dimensions are significantly high, the count is close to r for truly significant keywords.
4. By sorting the different keywords in decreasing order of their accumulated weights, the truly significant weights will be at the top of the sorted lists.
5. Therefore, if we want the k most important keywords per unit, we take the top k terms in the

sorted list that have greater than r° randomly projected dimensions that are significantly high. In our experiments, r°=0.6$*r$.

The pseudo-code of the procedure for extracting the significant keywords of a given unit q as described above is shown in Procedure 1. The vector *tallyFreq []* counts the number of times a term t has been tagged as significant (note that it can be tagged a maximum of r times, since each term is mapped to r dimensions). The vector *sumWeights []* accumulates the corresponding weights in the trained SOM, where w_{qd} is the d^{th} element of the weight vector of unit q. *RPM[][]* is the random projection matrix as described above. In the actual implementation of this algorithm, we have compressed the *RPM* matrix so that for each column, only the indices of the r dimensions for which the *RPM* matrix contains a 1 are stored. Also, in the search through the entries of the *sortedTermIndex []* vector of those words that have been tagged at least r° times, we only check the top T entries of the sorted list, as most of the numerous other terms are insignificant.

Figure 1 illustrates the relative distribution of significant keywords, piggy-back terms, and insignificant terms among the ordinary and significantly high dimensions. Significant keywords are mapped mainly to dimensions that have significantly high weights. This is a well-studied property of SOM weight vectors that tend towards the expected values of the individual weight components. Since the important keywords of a given text document are those words that appear relatively more frequently than the others, then the dimensions corresponding to these keywords will necessarily receive relatively higher component values. As for piggy-back terms, these are mapped mostly to 1 or 2 significant dimensions (and to r-2 or r-1 other ordinary dimensions). Insignificant terms (there are many of these) are not mapped to any significant dimension, and thus are mapped to r dimensions that are all ordinary. Since we accumulate the weight values of only those keywords that are mapped to significantly high dimensions, it is clear that insignificant terms get zero accumulated weights, while piggy-back terms will get less accumulated sum of weights than the truly significant keywords.

The keyword extraction technique by Lagus [9], against which our method will be benchmarked in section 5, does not use the weight vectors of the trained map. Their technique directly computes the relative frequencies of occurrence of all words in all the documents assigned to a given unit in the map. A goodness measure G, defined below, is used to rank the words as to how much they meaningfully represent a given unit:

$$G(w,j) = \left[\sum_{k \in A0j} F_k(w) \right] \frac{\sum_{k \in A0j} F_k(w)}{\sum_{k \in A0j} F_k(w) + \sum_{k \in A2j} F_k(w)} \quad (1)$$

where $A0_j$ is the region of units that form the same cluster as unit j and $A2_j$ is the region of units much farther away from unit j, considered to be outside the cluster. A unit k is in $A0_j$ if the map grid distance $d(k,j)$ is not greater than a parameter radius r_0. Unit k is in region $A2_j$ if $d(k,j)$ is not less than r_1. The region between $A0_j$ and $A2_j$ is termed as a "neutral zone" (greater than r_0 but less than r_1), and relative

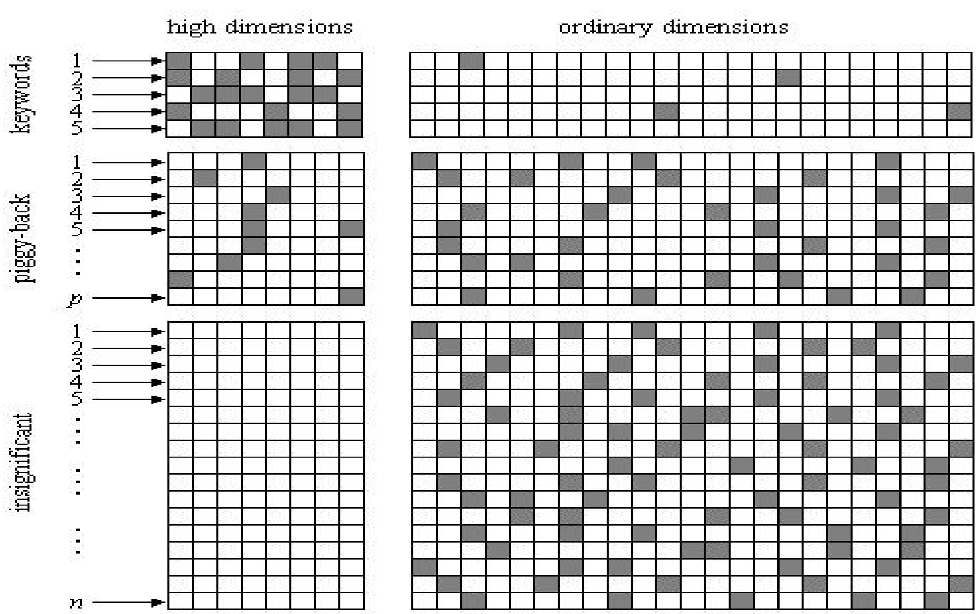

Figure 1. Significant keywords are mapped mainly to dimensions that have significantly high weights.

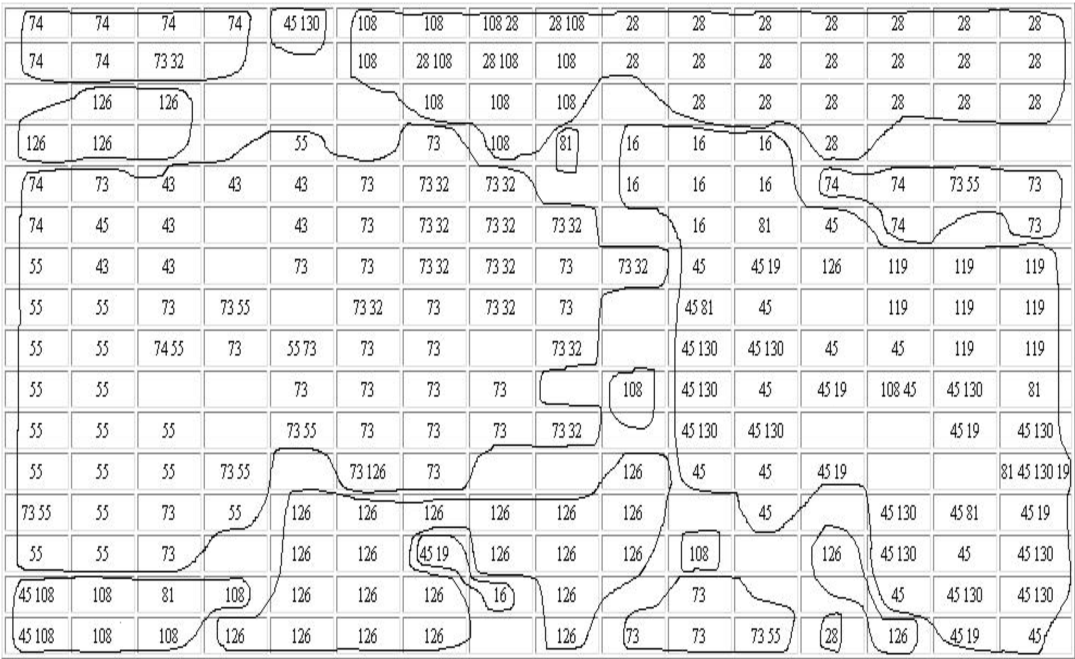

Figure 2. A 16x16 labeled SOM trained using the Reuters subset. Agricultural produce labels are: coffee (16), corn (19), grain (45), sugar (119), oil-seed (81) and wheat (130). Finance-related labels are: dollar (32), money-fx (73), money-supply (74), interest (55), and GNP (43). Other labels are trade (126), crude (28), and ship (108).

frequencies of words of the units in this region ($A1_j$) are not included in the computations.

The relative frequencies of each word w for a given unit k, denoted by $F_k(w)$, is defined in [9] as the number of times the term w occurs in unit k, denoted by $f_k(w)$, normalized by the total number of occurrences of all words in all the documents assigned to unit k. The relative frequencies are formally defined as follows (note that following the naming convention of [9], w stands for *word*, not *weight*):

$$F_k(w) = \frac{f_k(w)}{\sum_v f_k(v)} \tag{2}$$

Work done by Rauber and Merkl [4] [5] uses the weight vectors to find components (dimensions) that vary very little among all the documents assigned to the same unit. This is done by going back to all the documents assigned to a particular unit and computing the quantization error e_{ik} defined below, where w_{ik} is the k^{th} element of trained weight vector of the i^{th} map unit, and x_{jk} is the *tf x idf* entry of the k^{th} term for document j that is assigned to unit i.

$$e_{ik} = \sum_{x_j \in C_i} \sqrt{(w_{ik} - x_{jk})^2} \tag{3}$$

3. WEBSOM Archive of Reuters-21,578

The keyword deduction technique was applied to a WEBSOM 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. Several classification performance reports appear in the literature [15]-[18].

Once the WEBSOM is trained, each document is associated to a specific unit of the map (we refer to this process as "archiving"), and the cluster of documents associated to a given unit may have one or more labels. A given class label (e.g. *dollar, corn*) is assigned to a unit if at least 60% of the documents associated to the unit carries that label. Note that in the Reuters collection, each document has been manually assigned to one or more class labels.

The trained and labeled 16x16 SOM is shown in Figure 2. Observe that there is a clear grouping of units that are associated to news documents pertaining to *agricultural produce*, like *coffee, corn, grain, oilseed, sugar* and *wheat*. These are mainly grouped at the lower right hand section of the map. *Finance*-related news documents, e.g. *dollar, GNP, interest, money-fx*, and *money-supply* are grouped in the left half of the map. Furthermore, a small grouping of *ship* and *crude-oil* news documents is located on the upper

dlr fed week sai across loan	dlr week	dlr week borrow	dlr week		ship vessel	ship	ship gulf hormuz	tanker port		oil line	oil	oil	oil product well	oil	oil barrel reserv austrian
dlr billion	dlr billion	dlr	dlr share			wari disput		stockbrok	refineri	oil	oil	oil	oil price	oil price barrel	dlr oil price barrel
billion dlr	billion dlr	dlr billion					mt	redund	refineri ecuador	ecuador oil labor pipelin	oil ecuador labor	opec price bpd oil	price opec oil barrel	price oil	dlr crude price
billion februari	billion	billion year	year	credit unchang	govern				coffe produc quota	coffe produc ecuador colombia	coffe	price	price	reserv fed	dlr fed
billion januari	billion	year	year growth	year		dollar	dollar dealer	coffe	coffe quota produc ground seem ico	coffe export	coffe bag		reserv feder fed	reserv fed custom feder repurchas	fed dlr custom repurchas
pct	pct	pct	pct		dollar	dollar yen	dollar bui yen dealer	dollar dealer	coffe	coffe export	coffe	certif	reserv feder	reserv fed feder repurchas	fed custom feder repurchas
pct	pct	pct	pct		dollar	dollar central	dollar yen	dollar		rice			sugar	sugar	sugar cargo white
pct	pct	pct			exchang	dollar yen	dollar sai	sai				ec	sugar ec	sugar	sugar cost white equiti
pct	pct		bank	bank	exchang	exchang	sai baker	sai		wheat			ec	tonn sugar trader tradit	tonn sugar virtual white
pct bank prime		bank	bank	bank	exchang	exchang baker	baker treasuri	baker treasuri		offer wheat	wheat	maiz	tonn	tonn	tonn crusher virtual shipment
rate prime stick rais	bank rate prime	bank rate	bank	bank	currenc	currenc nation baker	baker	baker		wheat	crop area		tonn	tonn wheat export	tonn wheat
bank rate prime stick	rate bank stick	rate			taiwan			japanes	japanes	crop drought	crop	grain	tonn	tonn export	tonn wheat export corn
bank rate intervent franc market	rate bank stick cut	rate		taiwan	trade taiwan		trade	japanes	chip japanes		crop		tonn	tonn export	tonn corn report export
rate	market		futur contract	trade	trade	trade corn canadian	trade	trade japan japanes		stg	mln stg	mln		tonn export depart	tonn wheat export
	port union worker	strike	trade	trade	trade	trade countri	trade nil	stg	stg market mass	mln stg market	mln stg	mln	mln credit		tonn mln wheat
ship load grain	ship end strike	strike seamen brazil union	trade	trade	trade	trade volcker	trade	stg	stg market monei poehl revis mass todai	stg mln market mass	mln stg ration	mln	mln	mln	mln tonn

Figure 3. Top keywords per map unit. Note that extracted terms had been stemmed. This altered the spelling of some words, e.g. januari, monei, currenc, bui (buy), produc, coffe.

right-hand corner of the map, while a "*trade*" cluster is found at the lower middle section. A few other specialized clusters are also observed.

The deduction technique discussed in section 2 was applied on the Reuters map. We obviously expect the extracted keywords to coincide with the various labels that have been assigned to the various text documents, and in fact, augment these labels with keywords that describe better the cluster of documents that they represent. If we select the top keywords per unit at a **z**-value of 1.96, we would obtain the keyword table shown in Figure 3.

Comparing the keyword distribution of Figure 3 and the label distribution in Figure 2, we can see that the deduced keywords reflect very much the kind of map organization that emerged based on manually assigned category labels. For example, the units located in the large agricultural produce section which are labelled with “wheat”, “grain”, “corn”, “oil-seed”, and “sugar”, have such keywords as*: coffee*, *wheat, rice*, *sugar, product*, *grain*, *corn*, *maize*, *quota*, *Ecuador*, *Colombia*, *ground*, *seem*, *ico*, *export, bag, certify*, *cargo*, *white*, *ec*, *ton*, *equity*, *trader*, *tradition*, *crusher*, *virtual*, *shipment*, *area*, *crop*, *drought*, *depart*, *report*, *credit*, and *mln.*

The units located in the large finance section of the map which are labeled with “dollar”, “GNP”, “interest”, “money-fx”, and “money-supply”, have such keywords as: *dlr*, *dollar*, *yen*, *equity*, *bank*, *fed*, *loan*, *treasury*, *borrow*, *reserve*, *billion*, *February*, *share*, *price*, *credit*, *unchanged*, *govern*, *week*, *say*, *across*, *January*, *growth*, *dealer*, *pct*, *buy*, *exchange*, *Baker*, *prime*, *stick*, *raise*, *offer*, *rate*, *raise*, *currency*, *nation*, *trade*, *Taiwan*, *intervention*, *franc*, *market*, *cut*, *future*, and *contract.*

The extracted keywords also match the manually assigned labels of “crude” and “ship” at the upper right half section of the map. There is a small cluster of “ship”, “grain”, and “oil-seed” at the lower left-hand corner which seems oddly located. Upon inspection of its keywords, we see that the documents are in fact pertaining to news reports of various seaports in the world (e.g. in Brazil) where a union of seamen has staged a strike that has affected the trade of grain.

4. Comparing Label Extraction Techniques

To have a more methodical assessment of the list of keywords extracted by our method, we implemented the G measure discussed earlier (in Lagus [9], this is the G^2 measure) of the WEBSOM methodology and also extracted top keywords based on this measure. Over all, we can claim that our method extracts fairly the same keywords as what the Lagus method would extract by digging out all the words of all document of each and every unit.

Figure 4 presents % match rates of different radius combinations (recall that the Lagus method has two radius values as parameters) for a fixed **z**-value of 1.96. We found that r_0=1 gives the best match rates, although the combination r_0=0 and r_1=16 also gives fairly comparable match rates. Lagus [9] reports that r_1=5 gives the best match rates, although the assessment was made using a simpler G^1 measure.

Notice from Table 1 that depending on the **z**-value used, 56-73% of the top keywords extracted per unit using our method are also the top keywords for the same units using the Lagus method. 77-93% of the top keywords extracted using our method are among the top 3 keywords for the same units using the Lagus method. If we consider the top 3 keywords extracted per unit using our method, Table 2 shows that 50-82% of the top 3 keywords extracted using our method are also among the top 3 keywords for the same units using the Lagus method and 68-98% are among the top 8 keywords extracted for the same units using the Lagus method. We used r_0 = 1 and r_1 = 5 as parameters for the G^2 measure, which are typical values reported in [9].

Another interesting radius combination that can be gleaned from figure 4 is r_0=1 and r_1=16. This combination gives the highest % match rate with the Lagus method at z-value = 1.96. The G^2 measure based on this combination computes for the relative frequencies of all the words in

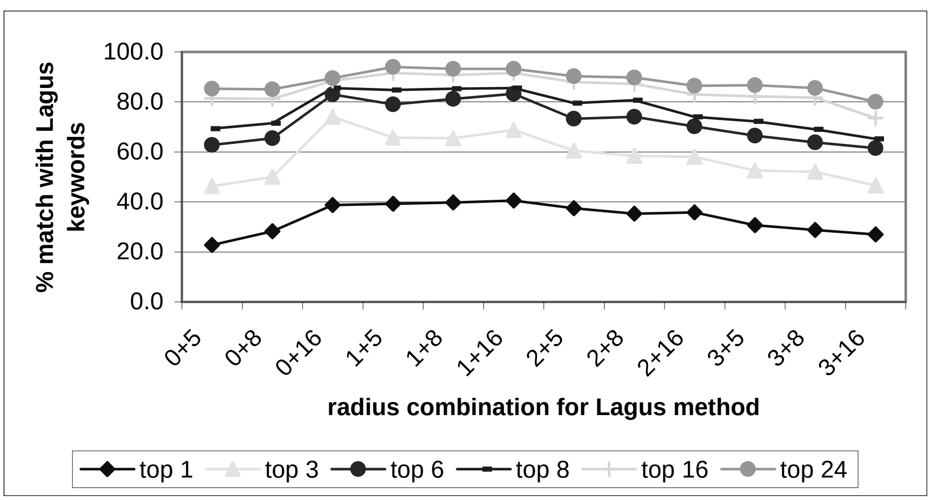

Figure 4 Percent of top 3 keywords per unit extracted that match the top k keywords (k=1, 3, 6, 8, 16, 24) extracted for the same units using the Lagus method with different radius combinations. A radius combination denoted 1+5 refers to r_0=1 and r_1=5. Best match rates are noted at r_0=1,

documents assigned to the given map unit plus the 8 other surrounding map units. All the other units in the map are ignored (neutral zone), because no two units in a 16x16 map can be more than 15 units apart (r_1=16). The G^2 measure penalizes words that appear in the cluster surrounding the given map unit if these words also appear in units outside the neutral zone. Our experiments with the Reuters archive indicate that our extraction method selects keywords regardless of whether the same keywords appear in documents associated to units much farther away in the map. Indeed, we may have units on opposite corners of the map that may have a common keyword. We differ from the Lagus [9] in this regard.

		% match with top *k* keywords							
z-value	# of keywords	1	2	3	4	5	6	7	8
1.282	251	**56**	71	**77**	81	84	85	86	71
1.645	235	**60**	74	**80**	83	86	87	87	74
1.960	211	**63**	76	**83**	88	91	92	92	76
2.326	157	**69**	79	**85**	90	92	94	94	79
2.576	130	**69**	82	**88**	93	95	96	96	82
3.090	88	**70**	83	**89**	94	97	97	98	83
3.291	74	**73**	86	**93**	97	97	97	97	86

Table 1 Percent of top keywords per unit extracted using our method that match the top k **keywords (**k**=1,2,…8) extracted for the same units using the Lagus method (using r_0 = 1 and r_1 = 5).**

		% match with top *k* keywords							
z-value	# of keywords	1	2	3	4	5	6	7	8
1.282	676	27	42	**50**	56	60	62	65	**68**
1.645	512	32	49	**58**	64	68	71	76	**79**
1.960	382	39	57	**66**	72	76	79	82	**85**
2.326	250	47	64	**71**	78	82	86	88	**91**
2.576	195	50	68	**74**	81	84	88	90	**93**
3.090	117	56	74	**81**	88	91	95	96	**98**
3.291	97	58	74	**82**	89	91	93	94	**96**

Table 2 Percent of top 3 keywords per unit extracted using our method that match the top k **keywords (**k**=1,2,…8) extracted for the same units using the Lagus method (using r_0 = 1 and r_1 = 5).**

It is a different matter altogether if a word is common to all units in the map. Neither Lagus' G measure nor our method will extract such words as keywords. In the Lagus method, this is done by explicitly penalizing such words through the use of r_1, since words appearing in units in the map that are of distance greater than r_1 will be counted towards the denominator of the second term of G^2. In our method, such words are not selected because the dimensions to which they are randomly projected will have high values for all the units where they appear and thus lose out in the $\mu+\mathbf{z}\sigma$ test. Only words that have associated weights that are significantly different from the mean (in a few units) will be selected. Our method does not check whether the units are located in contiguous locations in the map. However, the characteristics and properties of self-organizing maps would tend towards neighboring units having similar weight vectors, and hence, towards neighboring units having common "significant" keywords.

In our method, the number of keywords extracted per node depends on the **z**-value. Lower **z**-values yield many keywords, but not all of them may be truly meaningful. Keywords extracted using high **z**-values are all meaningful, but many units are left unlabelled. The Lagus method, on the other hand, extracts keywords for all map units and for any desired number of keywords per unit, which is only limited by the number of unique words in the cluster of documents associated to the unit. Depending on how it is looked at, our method's variable number of extracted keywords per map unit can be good or bad. It is good because we do not force labels on units if there are no meaningful labels among the documents associated to it. On the other hand, we can argue that a few not-so-meaningful labels are better than no labels at all. In the Lagus method, the labels are sorted according to their G^2 measure and as the user zooms in on the map, those with higher G^2 values are displayed earlier than those with lower values. This is a nice feature which we could adapt to our method, using the accumulated weights and the number of truly significant dimensions as bases for ranking keywords in their order of appearance during zooming in and out of the map.

5. Conclusion

A technique for deducing the most important keywords of each unit in a WEBSOM text archive is described. We demonstrate the effectiveness of our technique by applying it on a WEBSOM archive of the well known Reuters text collection. We demonstrate that the keywords extracted using our method are far more descriptive of the document clusters they label than the manually assigned class labels. We do a methodical assessment of the keywords extracted using our method by also implementing the G^2 measure used by the Kohonen's WEBSOM team in Helsinki and by comparing the results. A high percentage of the keywords we extract match the top keywords extracted for the same units using the Lagus method.

6. References

[1] X. Lin, D. Soergel, and G. Marchionini (1991). A Self-Organizing Semantic Map for Information Retrieval. In Proceedings of the International ACM SIGIR Conference on R&D in Information Retrieval. Chicago, Illinois.

[2] M. Dittenbach, D. Merkl, and A. Rauber (2000). Using Growing Hierarchical Self-Organizing Maps for Document Classification. European Symposium on Artificial Neural Networks, ESANN2000. Bruges, Belgium. April 26-28.

[3] D. Merkl and A. Rauber (2000). Uncovering the Hierarchical Structure of Text Archives by Using an Unsupervised Neural Networks with Adaptive Architecture. Pacific Asia Conference on Knowledge Discovery and Data Mining, PAKDD' 2000. Kyoto, Japan.

[4] A. Rauber and D. Merkl (1999). Automatic Labelling of Self-Organizing Maps: Making a Treasure Maps Reveal Its Secrets. In Proceedings of the 4th Pacific-Asia Conference on Knowledge Discovery and Data Mining, PAKDD99. Beijing, China.

[5] A. Rauber and D. Merkl (1999). Mining Text Archives: Creating Readable Maps to Structure and Describe Document Collections. PKDD99.

[6] T. Honkela, et al. (1997). WEBSOM-Self-Organizing Maps of Document Collections. Proceedings WSOM'97, Workshop on Self-Organizing Maps. Espoo, Finland. June 4-6.

[7] S. Kaski, et al. (1998). Statistical Aspects of the WEBSOM System in Organizing Document Collections. Computing Science and Statistics. Vol. 29. pp. 281-290.

[8] T. Kohonen, et al. (1999). Self-Organization of a Massive Document Collection. Kohonen Maps. Elsevier.

[9] Lagus et al. (1999). WEBSOM for Textual Data Mining. Artificial Intelligence Review. Vol. 13. pp. 345-364.

[10] T. Kohonen (1998). Self-Organization of Very Large Document Collections: State of the Art. International Conference on Artificial Neural Networks, ICANN98. Skovde, Sweden. September 2-4.

[11] D. W. Aha (1998) Feature weighting for lazy learning algorithms, In: H. Liu and H. Motoda (Eds.) Feature Extraction, Construction and Selection: A Data Mining Perspective. Norwell MA: Kluwer.

[12] E. H. (Sam) Han and G. Karypis (2000). Centroid-Based Document Classification: Analysis and Experiment Results. PKDD2000.

[13] S. Shankar and G. Karypis (2000). Weight Adjustment Schemes for a Centroid Based Classifier. Text Mining Workshop, KDD2000.

[14] D. Memmi, and J. G. Meunier (2000). Using competitive networks for text mining. Proceedings Neural Computation NC'2000. Berlin, Germany.

[15] Y. Yang (1999). An Evaluation of Statistical Approaches to Text Categorization. Journal of Information Retrieval. Vol. 1. No. 1/2. pp. 67-88.

[16] G. Salton (1989). Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer. Addison-Wesley, Reading, MA.

[17] H. T. Ng, W. B. Goh, and K. L. Low (1997). Feature Selection, Perceptron Learning, and a Usability Case Study for Text Categorization. SIGIR1997.

[18] A. Azcarraga, and T. Yap Jr. (2001). SOM-Based Methodology for Building Large Text Archives. 7th International Conference on Database Systems for Advanced Applications, DASFAA01. Hong Kong. April 18-20.