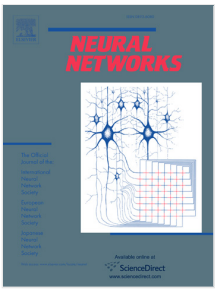
Accepted Manuscript



The Learning Vector Quantization algorithm applied to automatic text classification tasks

M.T. Martín-Valdivia, L.A. Ureña-López, M. García-Vega

PII: S0893-6080(07)00004-4
DOI: 10.1016/j.neunet.2006.12.005
Reference: NN 2252

To appear in: *Neural Networks*

Received date: 20 October 2005
Revised date: 12 December 2006
Accepted date: 12 December 2006

Please cite this article as: Martín-Valdivia, M. T., Ureña-López, L. A., & García-Vega, M. The Learning Vector Quantization algorithm applied to automatic text classification tasks. *Neural Networks* (2007), doi:10.1016/j.neunet.2006.12.005

The Learning Vector Quantization Algorithm Applied to Automatic Text Classification Tasks

M.T. Martín-Valdivia [1] L.A. Ureña-López
M. García-Vega

Department of Computing, University of Jaén, Campus Las Lagunillas s/n, Edificio A3, Jaén, E-23071 SPAIN.

[1] Corresponding author. Phone:+34.953.212898. Fax:+34.953.212472. E-mail: maite@ujaen.es.

The Learning Vector Quantization Algorithm Applied to Automatic Text Classification Tasks

Abstract

Automatic text classification is an important task for many natural language processing applications. This paper presents a neural approach to develop a text classifier based on the Learning Vector Quantization (LVQ) algorithm. The LVQ model is a classification method that uses a competitive supervised learning algorithm. The proposed method has been applied to two specific tasks: text categorization and word sense disambiguation. Experiments were carried out using the Reuters-21578 text collection (for text categorization) and the Senseval-3 corpus (for word sense disambiguation). The results obtained are very promising and show that our neural approach based on the LVQ algorithm is an alternative to other classification systems.

Keywords: Learning Vector Quantization (LVQ), Word Sense Disambiguation (WSD), Text Categorization (TC), SENSEVAL, Reuters-21578 text collection, Natural Language Processing (NLP)

1 Introduction

Document classification can be thought of as a problem of mapping the space between an input document and an output class. Neural networks can learn nonlinear mappings from a set of training patterns.

A Neural Network (NN) is an interconnected assembly of simple processing elements (called units or nodes), whose structure is inspired on animal neurons. Despite a large number of successful applications of NNs in a variety of areas (see [Rumelhart et al., 1994] for a survey of practical applications), their use in Natural Language Processing (NLP) tasks has not been explored sufficiently. In fact, there are no so many references as in other applications, for example in areas like optimization or pattern classification. However, NNs present various properties that NLP could take advantage of, such as massively parallel architecture, noise tolerance, self organization, and generalization. In fact, recently, interest in connecting both disciplines is growing spectacularly, as shown in [Dale et al., 2000].

In this paper, we discuss a neural classifier based on the Kohonen model which uses competitive supervised learning. In particular, we use the Learning Vector Quantization (LVQ) algorithm to accomplish two text classification tasks: Text Categorization (TC) and Word Sense Disambiguation (WSD).

The paper is organized as follows. First, we introduce briefly the automatic text classification task. Then, we describe the LVQ algorithm and the information representation model used in our experiments. After this, we show our evaluation environment and results for the two tasks considered (TC and WSD). Finally, we discuss our conclusion and future research.

2 Automatic Text Classification

Automatic text classification is one of the main tasks of NLP. Various approaches have been explored, such as support vector machine [Joachims, 1998], Naive Bayes learning methods [Lewis and Ringuette, 1994] or linear text classifiers [Lewis et al., 1996].

2.1 Neural Text Classification

Neural networks have also been applied to automatic text classification tasks. Certainly, the most widely used NN in NLP applications is the Back Propagation Network (BPN) proposed by [Rumelhart and McClelland, 1986]. For example, text classification has been investigated in several studies [Wiener et al., 1995, Ng et al., 1997, Yang and Liu, 1999, Wermter, 2000]. These papers propose different numbers of hidden layers and several architectures but all use the BPN model. However, other different neural models can be used, such as the Kohonen model [Kohonen, 1995]. Here we describe some examples.

[Lin, 1997] uses the unsupervised version of the Kohonen model (Self Organizing Map - SOM) to generate a classifier using a science document col-

lection (the corpus used is the LISA – *Library and Information Science Abstract* – text collection). The SOM model is also applied by [Chen et al., 1998] to classify and search Internet homepages according to their contents, and [Nakayama et al., 2000] automatically classify educational information. [Merkl, 1998] proposes a hierarchical structure combining multiple SOMs to design a classifier system. This model has been applied to several corpora and domains (software library [Merkl, 1993, Merkl et al., 1994], legal texts [Merkl et al., 1995], newspaper article collection [Dittenbach et al., 2001]...). The latest works by Merkl [Merkl and Rauber, 2000] are related to the SOMLib project[1] which is oriented to automatic classification of documents in digital libraries.

[Rauber and Merkl, 1999] make a comparison between the supervised (LVQ) and unsupervised (SOM) Kohonen models showing that the LVQ algorithm performs slightly better than the SOM model. Experiments were carried out using the TIME MAGAZINE collection[2]. [Goren-Bar et al., 2001] also compare LVQ and SOM using financial news from YAHOO and, again, the best results were obtained with the LVQ algorithm.

[Kohonen et al., 2000] describe the implementation of a system that is able to organize vast document collections according to textual similarities. It is based on the SOM algorithm. The system, called WEBSOM[3], is described in detail in [Honkela et al., 1996, Honkela et al., 1997, Honkela, 1997] but this architecture has also been used in other research [Lagus, 2002, Guerrero et al., 2002].

Recently, [Hung et al., 2004] integrate a guided SOM and other competitive neural learning with diverse knowledge sources, and [Hung and Wermter, 2004] use three different vector representation approaches extracting class knowledge for document classification.

Finally, our recent work [Martín-Valdivia et al., 2003] presents a neural model based on the LVQ algorithm to categorize a multilingual corpus (the polyglot Bible). The application of the LVQ algorithm to different NLP classification problems is described in [Martín-Valdivia, 2004].

2.2 Tasks

This paper presents the application of the LVQ algorithm to two automatic text classification tasks: Text Categorization and Word Sense Disambiguation.

- Text categorization is the task of assigning a Boolean value to each pair $< d_j, c_i > \in D \times C$, where D is a domain of documents and $C = c_1, ..., c_{|C|}$ is a set of predefined categories [Sebastiani, 2002]. A value 1 assigned to $< d_j, c_i >$ is interpreted as document d_j belongs to category c_i, while a value 0 indicates that document d_j does not belong to category c_i.

- Word sense disambiguation consists in identifying word meaning in a determined context [Kilgarriff and Palmer, 2000]. WSD can be seen as a

[1] http://www.ifs.tuwien.ac.at/ andi/somlib/
[2] http://www.ifs.tuwien.ac.at/~andi/somlib/experiments_time60.html
[3] http://websom.hut.fi/websom/

classification task where the categories are the different senses of a word and the documents are words and their contexts. Thus, the goal of a disambiguation system will be to assign for each word a label with the correct sense according to the word context.

Both cases require the use of a document collection labelled with classes (categories or senses). The collection needs to be divided into two subsets: a training subset to adjust the classifier, and a test subset to measure the effectiveness of the developed system.

3 LVQ Text Classification Approach

Typically, a text classifier based on a neural approach is a network of units, where the input units represent terms, the output units represent the categories of interest and the weight connections represent dependence relations. The neural classifier is trained by using a learning algorithm in order to modify and adjust the weights appropriately.

The Kohonen model [Kohonen, 1995] is one of the most widely used and referred to in the NN literature. The original Kohonen model is known as the Self Organizing Map (SOM), and is characterized by the formation of topological maps. The SOM uses competitive unsupervised learning based on the Winner-Takes-All (WTA) principle. In a competitive network, output units compete to classify input patterns.

The LVQ algorithm is the supervised version of the Kohonen model and it was especially designed to accomplish pattern classification tasks. In fact, most of the applications that use the LVQ algorithm are related to classification problems [Kohonen, 1995, Kaski et al., 1998].

3.1 The LVQ Algorithm

The LVQ algorithm [Kohonen, 1995] is notable for its heuristic simplicity and for directly adapting to text classification tasks. However, the application of the LVQ algorithm to text classification has not been sufficiently explored. The LVQ algorithm is a classification method based on neural competitive learning, which allows the definition of a group of categories on the input data space using reinforced learning, either positive (reward) or negative (punishment).

The neural architecture is quite simple. The Kohonen model does not include a layer of hidden units, so the neural network consists only of one input layer and one output layer. The input layer has as many units as features retained (i.e., dimensions of the input vectors) and there is one output unit for each possible class.

All the input units are fully-connected by feedforward connections to all output units. However, the output units are interconnected through lateral inhibitory connections so that, when an output unit is activated, it uses the lateral connections to send inhibition signals and to be able to deactivate the rest of the output units.

In the LVQ algorithm, weight vectors associated to each output unit are known as codebook vectors. Each class of input space is represented by its own set of codebook vectors. Codebook vectors are defined according to the specific task. For the categorization task we use one codebook vector per class. The category label for each codebook vector is the class name. For WSD we use as many codebook vectors as the specific word has senses to disambiguate. In this case, the category label for each codebook vector is the sense number of the word.

The LVQ algorithm is a competitive network, and thus, for each training vector, output units compete among themselves in order to find the winner according to some metric. The LVQ algorithm uses the Euclidean distance to find the winner unit. Only the winner unit (i.e., the output unit with the smallest Euclidean distance with regard to the input vector) will modify its weights using the LVQ learning rule. The basic LVQ algorithm is the following:

1. Initialize the codebook vectors W_i and the learning rate α
2. Randomly select an input vector X
3. Find the winner unit closest to the input vector (i.e., the codebook vector W_c with the smallest Euclidean distance with regard to the input vector X):
$$\|X - W_c\| = \min_k \|X - W_k\| \tag{1}$$
i.e.,
$$c = \arg\min_k \|X - W_k\| \tag{2}$$
4. Modify the weights of the winner unit:
   - If W_c and X belong to the same class (the classification has been correct)
   $$W_c(t+1) = W_c(t) + \alpha(t)[X(t) - W_c(t)] \tag{3}$$
   - If W_c and X belong to different classes (the classification has not been correct)
   $$W_c(t+1) = W_c(t) - \alpha(t)[X(t) - W_c(t)] \tag{4}$$
5. Reduce the learning rate α
6. Repeat from step 2 until the neural network is stabilized or until a fixed number of iterations have been carried out

The learning rate $\alpha(t)$ $(0 < \alpha(t) < 1)$ is a monotically decreasing function of time which controls how quickly the weight vector is allowed to change. It is recommended that $\alpha(t)$ should already initially be rather small, say, smaller than 0.3 and it continues decreasing to a given threshold very close to 0 [Kohonen, 1995].

3.2 Information Representation

In order to represent the document collection and categories appropriately, we use the Vector Space Model (VSM) as information representation scheme.

The VSM [Baeza-Yates and Ribiero-Neto, 1999] was originally developed for the Information Retrieval (IR) community, but it can be used in other NLP tasks such as TC or WSD [Manning and Schütze, 2000].

The VSM represents a document by a weighted vector of terms. A weight assigned to a term represents the relative importance of that term. One common approach for term weighting uses the frequency of occurrence of a particular word in the document to represent the vector components [Salton and McGill, 1983]. In order to calculate the term weights, we used the standard $tf \times idf$ equation, where tf is the frequency of the term in the document, and idf is the inverse document frequency defined as:

$$idf_i = \log_2 \left(\frac{M}{df_i} \right) \tag{5}$$

where df_i (document frequency) is the number of documents in the collection in which term i occurs, and M is the total number of documents.

Thus, weight w_{ij} is calculated by the following equation:

$$w_{ij} = tf_{ij} \times idf_i \tag{6}$$

where tf_{ij} (term frequency) is the number of occurrences of term i in document j.

In the VSM, categories are also represented by term weight vectors. To this end, a representative document for each category should be generated and the VSM for each category then is applied.

The similarity between documents and categories is computed by a distance measure, that is usually the cosine of the angle of their vectors. However, because we used the LVQ algorithm, the distance measure will be the Euclidean distance. Thus, the similarity between document j and category k is obtained with the following equation:

$$sim(d_j, c_k) = \|d_j - c_k\| = \sqrt{\sum_{i=1}^{N} (w_{ij} - c_{ik})^2} \tag{7}$$

where N is the number of terms in the whole collection, w_{ij} is the weight of term i in document j and c_{ik} is the weight of term i in category k.

4 Experiments

In this paper, we propose the use of the LVQ algorithm to train neural classifiers. The LVQ algorithm has been used to accomplish two particular tasks (TC and WSD). In both cases, the process is similar. We have a document collection labelled with categories. For the TC task, we used the REUTERS-21578 test

collection, and for the WSD case, the SENSEVAL-3 corpus and the linguistic resources SEMCOR and WORDNET.

The data collection is divided into two subsets: a training set and a test set. We use the vector space model to represent documents and categories by weighted vectors. To generate the category vectors we merged all documents belonging to one class and then we applied the VSM to all categories. Once we have represented the documents and categories with vectors, the training phase begins.

All input vectors are processed several times during training. In each iteration an input vector is randomly selected and compared with every weight vector using the Euclidean distance. The codebook vector closest to the input vector is the winner unit. The winner vector will modify its weights using the LVQ learning rule.

The test phase begins after the training. An input vector is selected and compared with every codebook vector. The closest codebook to the input vector is then selected as winner. The class assigned to the input vector is the class of the winner codebook vector.

4.1 Evaluation Measures

The effectiveness of an automatic classifier can be evaluated with several measures [Sebastiani, 2002]. The classical 'Precision' and 'Recall' for IR are adapted to the case of text classification. To that end, a contingency table for each category should be generated (Table 1), and the precision and recall for each category are then calculated with the following equations:

$$P_i = \frac{a_i}{a_i + b_i} \tag{8}$$

$$R_i = \frac{a_i}{a_i + c_i} \tag{9}$$

In order to measure the average performance of a classifier over all the categories, two measures can be used: micro-averaging precision P_μ and macro-averaging precision P_{macro}.

$$P_\mu = \frac{\sum_{i=1}^{K} a_i}{\sum_{i=1}^{K} (a_i + c_i)} \tag{10}$$

$$P_{macro} = \frac{\sum_{i=1}^{K} P_i}{K} \tag{11}$$

where K is the number of categories.

4.2 LVQ applied to Text Categorization

TC is the classification of documents according to a set of one or more pre-existing categories. TC is a difficult but useful operation frequently applied to

the assignment of subject categories to documents, to route and filter texts, or as a part of natural language processing systems [Lewis, 1992].

The LVQ algorithm has been used to automatically categorize the REUTERS test collection[4] according to its contents. REUTERS is a linguistic resource widely used in TC [Lewis, 1992] with the purpose of measuring the effectiveness of TC systems. The REUTERS-21578 collection consists of 21,578 newswire stories about financial categories collected from REUTERS during 1987. For each document, a human indexer decided which categories from which sets that document belonged to. There are 135 different categories, which are overlapping and non-exhaustive, and there are relationships among the categories. Figure 1 shows a document from REUTERS-21578 with TOPICS categories 'crude' and 'nat-gas'.

The REUTERS collection can be divided into various training and test subsets. One of the most popular partitions is the MODAPTE Split [Apte et al., 1994]. In the MODAPTE partition, all documents are assigned at least one category. Our experiments have been carried out with this division (12,902 documents: 9,603 for training and 3,299 for test), but we have selected only documents with exactly one category (overall, 11,216 documents: 8,355 for training and 2,861 for test).

Firstly, the MODAPTE partition has been pre-processed as usual, removing common words with the SMART[5] stoplist and extracting the word stems using the Porter algorithm [Baeza-Yates and Ribiero-Neto, 1999]. After preprocessing, a total of 23,140 different terms and 60 different categories were found. Thus, the neural architecture is 23,140 input units and 60 output units.

The dimension of input vectors (documents) and codebook vectors (categories) is 23,140. To generate the input vector, the VSM is applied to each document. In order to generate the vectors for each category, we merged all documents belonging to a category and then applied the VSM.

Once we have represented the documents and categories with vectors, the training phase begins. All training vectors were presented to the neural network several times and the learning rate $\alpha(t)$ was initialized to 0.3. The neural classifier has been trained with the proposed LVQ algorithm using the training collection and then we evaluated the effectiveness with the test collection.

In order to compare the results obtained, we applied the Rocchio algorithm [Rocchio, 1971]. The Rocchio algorithm is one of the most widely-used algorithms in text categorization and it serves as baseline case in many other studies [Sebastiani, 2002]. The model is adjusted using the implementation described in [Lewis et al., 1996].

The results obtained with the LVQ and the Rocchio algorithms are shown in Table 2. As can be seen, the LVQ algorithm performs significantly better than the Rocchio algorithm. The macro-averaging and the micro-averaging precision

[4]The REUTERS-21578 text categorization test collection is available at http://www.daviddlewis.com/resources/testcollections/reuters21578/, thanks to REUTERS, Carnegie Group, and David Lewis.

[5]SMART is one of the best known IR experimental systems of public domain at ftp://ftp.cs.cornell.edu/pub/smart

improvement is 19.61% and 23.73%, respectively.

With the LVQ algorithm, the best result is obtained for the 'earn' category and the worst result is obtained for the 'coffee' category. Table 3 and Table 4 show the contingency table for these two cases. The precision for the 'earn' category is 0.80 with recall 0.97. However, we obtained a precision of 0.10 and recall 0.96 for the worst case ('coffee' category).

On the other hand, the comparison of results of the LVQ algorithm with other classifiers is quite complex, as shown in [Yang, 1999], even using the same collection. Yang demonstrates that the experiments with the REUTERS collection but with different partitions are not directly comparable since the conditions are different. For this reason, the comparison of the LVQ algorithm with other TC system will be accomplished in an indirect form considering solely the percentage of improvement of the Rocchio algorithm applied on the REUTERS collection.

Thus for example[6], [Schütze et al., 1995] uses two network architectures that improve the Rocchio algorithm around 15%. One of the networks includes a hidden layer while the other does not, but both systems use a back propagation algorithm. [Joachims, 1998] makes a comparative study of several TC systems including the Rocchio algorithm, verifying that a bayesian classifier and another one based on decision trees give worse results than the Rocchio algorithm, whereas the methods based on k-NN (*k-Nearest Neighbors*) and on SVM (*Support Vector Machine*) perform slightly better (the improvement is about 8%). [Li and Yamanishi, 1999] present a comparison with decision lists, SVM, Naive Bayes and k-NN but the improvement on Rocchio is never above 9%. [Lai and Lam, 2001] compare the Rocchio algorithm with a system based on the Widrow-Hoff scheme [Widrow and Winter, 1988] obtaining an improvement of 16%. Also, [Ureña López, 2002] makes several comparisons between Rocchio and Widrow-Hoff using the integration of linguistic resources but the improvement obtained is about 9%. Finally, [Eyheramendy et al., 2003] use the previous version of the REUTERS collection with a TC based on k-NN and another one based on SVM. The results show that the former improved on Rocchio by 11%, whereas the latter did by almost 19%. Although all these methods obtain better results than the Rocchio algorithm, the improvement of LVQ over Rocchio is better than all theses systems.

4.3 LVQ applied to Word Sense Disambiguation

WSD consists of identifying word meaning in a certain context. The study of WSD is challenging, because the research outcome would highly influence the performance of many other NLP tasks. In fact, WSD is considered an 'intermediate task' [Wilks and Stevenson, 1998] which is not an end in itself, but which is necessary at one level or another to accomplish most NLP tasks, like information retrieval [Schütze, 1998], machine translation [Brown et al., 1991], text categorization [Ureña-López et al., 2001], dialogue system and information extraction

[6]The following examples only deal with documents which are assigned to exactly one category.

[Kilgarriff, 1997]. For example, an English–Spanish machine translation system needs to know when the word 'car' is being used to mean 'automobile', 'railway car' or 'elevator car' to decide whether it should be translated as 'coche', 'vagón' or 'ascensor'.

The LVQ algorithm has been used to accomplish the word sense disambiguation task. In our experiments, we used the SENSEVAL-3 English corpus. SENSEVAL[7] [Kilgarriff, 1998] is an international meeting whose purpose is to organize a competition to evaluate the strengths and weaknesses of WSD systems with respect to different words, different language, and different tasks.

In order to improve the training corpus made available by the SENSEVAL-3 organization, we integrated semantic information from two linguistic resources: the SEMCOR 1.6 corpus [Miller et al., 1993] and the WORDNET 1.7.1 lexical database [Fellbaum, 1998].

Firstly, the SEMCOR (the Brown Corpus labelled with the WORDNET senses) was treated in full (the Brown-1, Brown-2 and Brown-v partitions). We used the paragraph as a contextual semantic unit and each context was included in the training vector set. Figure 2 shows an extract of the br-f03 SEMCOR document with an occurrence of the word 'car'. The SENSEVAL-3 English tasks used the WORDNET 1.7.1 sense inventory, but the SEMCOR is tagged with an earlier version of WORDNET (specifically WORDNET version 1.6). Therefore, it was necessary to update the SEMCOR word senses. We used our automatically mapped version of SEMCOR with the WORDNET 1.7.1 senses found in the WORDNET site[8].

From WORDNET 1.7.1 some semantic relations were considered, specifically, synonymy, antonymy, hyponymy, homonymy, hyperonymy, meronymy, and coordinate terms. This information was introduced to the training set through the creation of artificial paragraphs with the words of each relation. For a word with 5 senses, 5 artificial paragraphs with the synonyms of the 5 senses were added, 5 more with all its hyponyms, and so on. Figure 3 shows the 5 artificial paragraphs for the 5 synonyms of 'car'.

Figure 4 shows the common format for the context generated with the linguistic resources. For each word, the POS (Part-Of-Speech) and sense are shown, e.g. 'car\1#1' is the noun 'car' with sense 1. In addition, there are 112 different words in its context and all of them are shown as word-frequency pairs.

The new training corpus generated with SEMCOR and WORDNET was used to train the WSD system. Every context of every word to disambiguate constitutes a domain. Each domain represents a word and its senses. We generated one network per domain and, after the training process, we have as many networks as domains/words to disambiguate with their adjusted weights. The network architecture per domain is shown in Figure 5. The number of input units is the number of different terms in all contexts of the given domain, and the number of output units is the number of different senses.

[7] http://www.senseval.org/
[8] http://www.cogsci.princeton.edu/~wn/

As in the text categorization task, the proposed disambiguator uses the vector space model as an information representation model. For each network, the input documents are the different contexts of a word, and the codebook vectors (i.e., the categories) are the different word senses. In order to initialize the codebook vectors we merged all contexts belonging to one sense and then we applied the VSM to all senses. Thus, each sense of a word is represented as a vector in an n-dimensional space where n is the number of words in all their contexts.

Once we have represented documents/contexts and categories/senses with vectors, the training phase begins. We use the LVQ algorithm to adjust the weight vectors. All training vectors were presented to neural networks several times and the learning rate $\alpha(t)$ was initialized to 0.3.

In each iteration, an input vector is selected and compared with every weight vector using the Euclidean distance, so that the codebook vector closest to the input vector wins. Only the winner class/sense will modify its weights using the LVQ learning rule. Thus, if the winner class and the input vector have the same class (i.e., the classification has been correct), it will increase its weights, coming slightly closer to the input vector. Otherwise, if the winner class is different from the input vector class (i.e., the classification has not been correct), it will decrease its weights, moving slightly further from the input vector.

After training, the codebook vectors contain the adjusted weights for all senses of each word, and the test phase begins. An input vector is selected and compared with every codebook vector. Then the closest codebook to the input vector is selected as winner. The class/sense assigned to the input vector (i.e., the disambiguated sense) will be the class/sense of the winner codebook vector. If it is not possible to find a sense, we assign by default the most frequent sense (i.e., the first sense in WordNet). Figure 6 shows the codebook vectors after training the word 'car'.

We participated at the Senseval-3 competition which took place in March 2004 [García-Vega et al., 2004]. The LVQ disambiguator was applied in two English tasks: English-Lexical-Sample (ELS) and English-All-Words (EAW). The main difference between both tasks is that the disambiguation of a limited set of words is required for ELS, whereas for EAW the system must disambiguate the whole text.

In order to train the neural networks, we used the available Senseval-3 corpus integrating the linguistic resources SemCor and WordNet. For the ELS task, we used only the contexts generated using SemCor and WordNet for each word at Senseval-3 corpus. A neural network was needed for each different word occurring at the ELS test. Finally, 57 neural networks for the 57 domains/words were trained using 8,215 input vectors. For the EAW task, we integrated the complete contexts of both linguistic resources. WordNet has a great number of fine-grained word sense distinctions. Many of the senses are very similar and they can be merged to yield a new coarse-grained classification. Both cases are considered at Senseval competition. A neural network is also needed for each different word present at the EAW evaluation text. In this case, we trained 982 neural networks using 79,042 input vectors, 80 vectors per

domain average.

The official results achieved by the WSD system based on the LVQ algorithm are presented in Table 5 for the ELS task [Mihalcea et al., 2004], and in Table 6 for EAW [Snyder and Palmer, 2004]. Of the 47 systems submitted to the ELS task, our system was 32nd, and of the 26 submitted to the EAW task, our system ranked 9th. Tables also show the best and worst results obtained at SENSEVAL-3.

For the ELS task, this edition of SENSEVAL showed a predominance of kernel-based methods (e.g. SVM) which were used by most of the systems. For example, the best system (Htsa3) uses Regularized Least-Squares Classification (RLSC) as learning method, which is based on kernels and Tikhonov Regularization. The second system (ITC-IRST) works with the kernel function to integrate diverse knowledge sources. However, our system is the only one based on a neural approach in SENSEVAL-3.

Regarding the EAW task, the two best systems apply Memory Based Learning (MBL) using TiMBL, but there are significant differences in the performance and the approaches of the systems.

5 Conclusions and Future Work

This paper presents a neural approach to automatic text classification, specifically, we use the LVQ algorithm. This neural network is a supervised learning algorithm based on the Kohonen model and we use it to automatic classify a document collection according to its content.

The proposed method has been applied to accomplish two different but closely related tasks: Firstly, we use the LVQ algorithm to categorize the REUTERS-21578 text collection; The second experiment has been carried out using the SENSEVAL-3 corpus in order to generate a disambiguator system based on the LVQ algorithm. The experiments show that the LVQ model performs successfully in both tasks.

The results obtained encourage us to continue working with the LVQ algorithm in other NLP tasks such as named entity classification, automatic summary generation or question and answering systems. We intend to apply the SOM model to carry out unsupervised text classification tasks.

Acknowledgements

This work has been supported by the Spanish Government (MCYT) with grant FIT-150500-2003-412.

References

[Apte et al., 1994] Apte, C., Damerau, F., and Weiss, S. (1994). Automated learning of decision rules for text categorization. *Information Systems*,

12(3):233–251.

[Baeza-Yates and Ribiero-Neto, 1999] Baeza-Yates, R. and Ribiero-Neto, B. (1999). *Modern Information Retrieval*. Addison-Wesley.

[Brown et al., 1991] Brown, P., Della-Pietra, S., Della-Pietra, V., and Merce, J. (1991). Word sense disambiguation using statistical methods. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL-91)*.

[Chen et al., 1998] Chen, H., Houston, A., Sewell, R., and Schatz, B. (1998). Internet browsing and searching: User evaluations of category map and concept space techniques. *Journal of the American Society for Information Science*, 49:582–603.

[Dale et al., 2000] Dale, R., Moisl, H., and Somers, H. (2000). *Handbook of Natural Language Processing*. Marcel Dekker, Inc., New York.

[Dittenbach et al., 2001] Dittenbach, M., Merkl, D., and Rauber, A. (2001). Hierarchical clustering of document archives with the growing hierarchical self-organizing map. In *Proceedings of International Conference on Artificial Neural Networks (ICANN-01)*, pages 500–505.

[Eyheramendy et al., 2003] Eyheramendy, S., Genkin, A., Ju, W., Lewis, D., and Madigan, D. (2003). Sparse bayesian classifiers for text categorization. Technical report, DIMACS Working Group on Monitoring Message Streams.

[Fellbaum, 1998] Fellbaum, C. (1998). WordNet: An electronic lexical database. *The MIT Press*.

[García-Vega et al., 2004] García-Vega, M., García-Cumbreras, M., Martín-Valdivia, M., and Ureña López, L. (2004). The University of Jaen word sense disambiguation system. In Mihalcea, R. and Edmonds, P., editors, *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 121–124.

[Goren-Bar et al., 2001] Goren-Bar, D., Kuflik, T., Lev, D., and Shoval, P. (2001). Automating personal categorization using artificial neural networks. In *Proceedings of User Modeling 2001*, pages 188–198.

[Guerrero et al., 2002] Guerrero, V., Moya, F., and Herrero, V. (2002). Document organization using Kohonen's algorithm. *Information Processing and Management*, 38:79–89.

[Honkela, 1997] Honkela, T. (1997). *Self-Organizing Maps in Natural Language Processing*. PhD thesis, Helsinki University.

[Honkela et al., 1996] Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1996). Newsgroup exploration with WEBSOM method and browsing interface. Technical report, Helsinki University of Technology, Espoo, Finland.

[Honkela et al., 1997] Honkela, T., Kaski, S., Lagus, K., and Kohonen, T. (1997). WEBSOM self-organizing maps of document collections. In *Workshop on Self-Organizing Maps*, pages 310–315, Espoo, Finland.

[Hung and Wermter, 2004] Hung, C. and Wermter, S. (2004). Neural network based document clustering using wordnet ontologies. *International Journal of Hybrid Intelligent Systems*, 1(3):127–142.

[Hung et al., 2004] Hung, C., Wermter, S., and Smith, P. (2004). Hybrid neural document clustering using guided self-organization and wordnet. *IEEE-Intelligent Systems*, 19(2):68–77.

[Joachims, 1998] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of European Conference on Machine Learning (ECML-98)*, pages 137–142.

[Kaski et al., 1998] Kaski, S., Kangas, J., and Kohonen, T. (1998). Bibliography of self-organizing map (SOM) papers: 1981-1997. *Neural Computing Surveys*, 1:1–176.

[Kilgarriff, 1997] Kilgarriff, A. (1997). What is word sense disambiguation good for? In *Proceedings of Natural Language Processing Pacific Rim Symposium.*

[Kilgarriff, 1998] Kilgarriff, A. (1998). SENSEVAL: An exercise in evaluating word sense disambiguation programs. In *Proceedings of International Conference on Language Resources and Evaluation (LREC-98).*

[Kilgarriff and Palmer, 2000] Kilgarriff, A. and Palmer, M. (2000). Introduction to the special issue on SENSEVAL. *Computers and the Humanities*, 24:1–13.

[Kohonen, 1995] Kohonen, T. (1995). *Self-organization and associative memory.* Springer-Verlag, Berlin.

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

[Lagus, 2002] Lagus, K. (2002). Text retrieval using self-organized document maps. *Neural Processing Letters*, 15:21–29.

[Lai and Lam, 2001] Lai, K. and Lam, W. (2001). Automatic textual document categorization using multiple similarity-based models. In *Proceedings of the 1st Siam International Conference on Data Mining (SDM-01).*

[Lewis, 1992] Lewis, D. (1992). *Representation and learning in information retrieval.* PhD thesis, Department of Computer and Information Science, University of Massachusetts.

[Lewis and Ringuette, 1994] Lewis, D. and Ringuette, M. (1994). Comparison of two learning algorithms for text categorization. In *Proceedings of the 3rd Annual Symposium on Document Analysis and Information Retrieval (SDAIR-94)*.

[Lewis et al., 1996] Lewis, D., Schapire, R., Callan, J., and Papka, R. (1996). Training algorithms for linear text classifiers. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-96)*.

[Li and Yamanishi, 1999] Li, H. and Yamanishi, K. (1999). Text classification using ESC-based stochastic decision lists. In *Proceedings of the 8th ACM International Conference on Information and Knowledge Management (CIKM-99)*, pages 122–130, Kansas City, US. ACM Press, New York, US.

[Lin, 1997] Lin, X. (1997). Maps displays for information retrieval. *Journal of the American Society for Information Science*, 48:40–54.

[Manning and Schütze, 2000] Manning, C. and Schütze, H. (2000). *Foundations of statistical natural language processing*. The MIT Press.

[Martín-Valdivia, 2004] Martín-Valdivia, M. (2004). *Algoritmo LVQ aplicado a tareas del procesamiento del lenguaje natural*. PhD thesis, Department of Computer Science, University of Málaga, Spain.

[Martín-Valdivia et al., 2003] Martín-Valdivia, M., García-Vega, M., and Ureña López, L. (2003). LVQ for text categorization using multilingual linguistic resource. *Neurocomputing*, 55:665–679.

[Merkl, 1993] Merkl, D. (1993). Structuring software for reuse—the case of self-organizing maps. In *Proceedings of International Joint Conference on Neural Networks (IJCNN-93), Nagoya*, volume III, pages 2468–2471, Piscataway, NJ. JNNS, IEEE Service Center.

[Merkl, 1998] Merkl, D. (1998). Text classification with self-organizing maps: some lessons learned. *Neurocomputing*, 21(1):61–77.

[Merkl and Rauber, 2000] Merkl, D. and Rauber, A. (2000). Digital libraries-classification and visualization techniques. In *Proceedings of International Conference on Digital Libraries: Research and Practice*, pages 434–438.

[Merkl et al., 1995] Merkl, D., Schweighofer, E., and Winiwater, W. (1995). Analysis of legal thesauri based on self-organising feature maps. In *Proceedings of the 4th International Conference on Artificial Neural Networks (ICANN-95)*, pages 29–34, London, UK. Vienna University of Technology, Austria, IEE.

[Merkl et al., 1994] Merkl, D., Tjoa, A., and Kappel, G. (1994). Application of self-organizing feature maps with lateral inhibition to structure a library of reusable software components. In *Proceedings of International Conference on*

Neural Networks (ICNN-94), pages 3905–3908, Piscataway, NJ. IEEE Service Center.

[Mihalcea et al., 2004] Mihalcea, R., Chklovski, T., and Kilgarriff, A. (2004). The Senseval-3 English lexical sample task. In Mihalcea, R. and Edmonds, P., editors, *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 25–28.

[Miller et al., 1993] Miller, G., Leacock, C., Randee, T., and Bunker, R. (1993). A semantic concordance. In *DARPA Workshop on Human Language Technology*.

[Nakayama et al., 2000] Nakayama, M., Sanematsu, H., and Shimizu, Y. (2000). A document indexing and retrieval method based on a teaching guideline for keyword searching educational information. *Transactions of the Institute of Electronics, Information and Communication Engineers*, pages 225–33.

[Ng et al., 1997] Ng, H., Goh, W., and Low, K. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. In *Proceedings of the 20st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-97)*, pages 67–73.

[Rauber and Merkl, 1999] Rauber, A. and Merkl, D. (1999). Using self-organizing maps to organize document archives and to characterize subject matter: how to make a map tell the news of the world. In *Proceedings of the 10th International Conference Database and Expert Systems Applications (DEXA-99). Lecture Notes in Computer Science Vol. 1677*, pages 302–311, Berlin, Germany. Springer-Verlag.

[Rocchio, 1971] Rocchio, J. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*, chapter Relevance feedback in information retrieval. Prentice Hall.

[Rumelhart and McClelland, 1986] Rumelhart, D. and McClelland, J. (1986). *Parallel Distributed Processing*, volume I+II. The MIT Press.

[Rumelhart et al., 1994] Rumelhart, D., Widrow, B., and Lehr, M. (1994). The basic ideas in neural networks. *Communications of the ACM*, 37(3):87–92.

[Salton and McGill, 1983] Salton, G. and McGill, M. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, London, U.K.

[Schütze, 1998] Schütze, H. (1998). Automatic word sense discrimination. *Computational Linguistics*, 24(1).

[Schütze et al., 1995] Schütze, H., Hull, D., and Pedersen, J. (1995). A comparison of classifiers and document representations for the routing problem. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-95)*, pages 229–237, New York. The ACM Press.

[Sebastiani, 2002] Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

[Snyder and Palmer, 2004] Snyder, B. and Palmer, M. (2004). The English all-words task. In Mihalcea, R. and Edmonds, P., editors, *Proceedings of Senseval-3: The Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43.

[Ureña López, 2002] Ureña López, L. (2002). *Resolución de la ambigüedad léxica en tareas de clasificación automática de documentos*. Colección de Monografías de la Sociedad Española para el Procesamiento del Lenguaje Natural.

[Ureña-López et al., 2001] Ureña-López, L., Buenaga-Rodríguez, M., and Gómez-Hidalgo, J. (2001). Integrating linguistic resources in TC through WSD. *Computer and the Humanities*, 35:215–230.

[Wermter, 2000] Wermter, S. (2000). Neural network agents for learning semantic text clasification. *Information Retrieval*, 3:87–103.

[Widrow and Winter, 1988] Widrow, B. and Winter, R. (1988). Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Computer*, 21:25–39.

[Wiener et al., 1995] Wiener, E., Pedersen, J., and Weigend, A. (1995). A neural network approach to topic spotting. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*.

[Wilks and Stevenson, 1998] Wilks, Y. and Stevenson, M. (1998). Word sense disambiguation using optimised combinations of knowledge sources. In *Proceedings of the 17th International Conference on Computational Linguistics and 36th Annual Meeting of the Association for Computational Linguistics (COLING-ACL-98)*.

[Yang, 1999] Yang, Y. (1999). An evaluation of statistical approaches to text categorization. *Information Retrieval Journal*, 1:69–90.

[Yang and Liu, 1999] Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR-99)*.

```
<REUTERS TOPICS=‘‘YES’’ LEWISSPLIT=‘‘TRAIN’’
CGISPLIT=‘‘TRAINING-SET’’ OLDID=‘‘18425’’ NEWID=‘‘2007’’>
<DATE> 5-MAR-1987 09:24:40.64</DATE>
<TOPICS><D>crude</D><D>nat-gas</D></TOPICS>
<PLACES><D>canada</D></PLACES> <PEOPLE></PEOPLE>
<ORGS></ORGS><EXCHANGES></EXCHANGES><COMPANIES></COMPANIES>
<UNKNOWN>&#5;&#5;&#5;E F Y &#22;&#22;&#1;f0025&#31;reuter
f BC-orbit-oil-increases   03-05 0094</UNKNOWN> <TEXT>&#2;
<TITLE>ORBIT INCREASES OIL AND GAS RESERVE VALUES</TITLE>
<DATELINE>    CALGARY, Alberta, March 5 - </DATELINE>
<BODY>&lt;Orbit Oil and Gas Ltd> said the value of its
 oil and gas reserves increased by 19 pct to 52.6 mln
dlrs from 44.2 mln dlrs reported at year-end 1985,
according to an independent appraisal. Orbit said it
has reserves of 2.4 mln barrels of oil and natural gas
liquids and 67.2 billion cubic feet of natural gas. In
addition, 75 pct owned &lt;Sienna Resources Ltd> has
Canadian reserves of 173,000 barrels of oil and 1.6
bcf of natural gas with a current value of 2.2 mln dlrs,
Orbit said. Reuter&#3;</BODY></TEXT> </REUTERS>
```

Figure 1: Document number 2,007 from REUTERS-21578

```
<s snum=28>
<wf cmd=done pos=RB lemma=for_instance wnsn=1 lexsn=4:02:00::>For_instance</wf>
<punc>,</punc>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=NN lemma=dreamer wnsn=1 lexsn=1:18:02::>dreamer</wf>
<wf cmd=done pos=VB lemma=see wnsn=4 lexsn=2:36:00::>sees</wf>
<wf cmd=ignore pos=PRP>himself</wf>
<wf cmd=done pos=VB lemma=seat wnsn=1 lexsn=2:35:00::>seated</wf>
<wf cmd=ignore pos=IN>behind</wf>
<wf cmd=done pos=NN lemma=neighbor wnsn=1 lexsn=1:18:00::>neighbor</wf>
<wf cmd=done rdf=person pos=NNP lemma=person wnsn=1 lexsn=1:03:00:: pn=person>Smith</wf>
<wf cmd=ignore pos=CC>and</wf>
<punc>,</punc>
<wf cmd=ignore pos=IN>with</wf>
<wf cmd=done pos=JJ lemma=photographic wnsn=1 lexsn=3:01:00::>photographic</wf>
<wf cmd=done pos=NN lemma=realism wnsn=1 lexsn=1:07:00::>realism</wf>
<punc>,</punc>
<wf cmd=done pos=VB lemma=see wnsn=4 lexsn=2:36:00::>sees</wf>
<wf cmd=done rdf=person pos=NNP lemma=person wnsn=1 lexsn=1:03:00:: pn=person>Smith</wf>
<wf cmd=done pos=VB lemma=drive wnsn=1 lexsn=2:38:01::>driving</wf>
<wf cmd=ignore pos=DT>the</wf>
<wf cmd=done pos=NN lemma=car wnsn=1 lexsn=1:06:00::>car</wf>
<punc>;</punc>
<wf cmd=ignore pos=IN>whereas</wf>
<punc>,</punc>
<wf cmd=ignore pos=PRP>it</wf>
<wf cmd=done pos=VBZ ot=notag>is</wf>
<wf cmd=ignore pos=DT>a</wf>
<wf cmd=done pos=NN lemma=matter wnsn=2 lexsn=1:09:01::>matter</wf>
<wf cmd=ignore pos=IN>of</wf>
<wf cmd=done pos=NN lemma=fact wnsn=1 lexsn=1:09:01::>fact</wf>
<wf cmd=ignore pos=IN>that</wf>
<wf cmd=done rdf=person pos=NNP lemma=person wnsn=1 lexsn=1:03:00:: pn=person>Smith</wf>
<wf cmd=ignore pos=MD>cannot</wf>
<wf cmd=done pos=VB lemma=drive wnsn=1 lexsn=2:38:01::>drive</wf>
<wf cmd=ignore pos=DT>a</wf>
<wf cmd=done pos=NN lemma=car wnsn=1 lexsn=1:06:00::>car</wf>
<punc>.</punc>
</s>
```

Figure 2: Extract of an occurrence of the word 'car' in br-f03 SemCor document

```
car\1#1 4 auto\1#1 1 automobile\1#1 1 machine\1#4 1 motorcar\1#1 1
car\1#2 3 railcar\1#1 1 railway_car\1#1 1 railroad_car\1#1 1
car\1#3 1 cable_car\1#1 1
car\1#4 1 gondola\1#3 1
car\1#5 1 elevator_car\1#1 1
```

Figure 3: Artificial paragraphs added from WordNet synonyms of 'car'

```
car\1#1 112 acre\1#1 1 affection\1#1 1 aggressive\5#2 1 ...
urban\3#2 1 western\5#2 1 worker\1#1 1

car\1#1 119 afford\2#1 1 all_of\3#1 1 always\4#1 2 ... stop\2#1
1 thumb\2#1 1
...
car\1#2 12 by\4#1 1 call\2#5 1 ... wink\2#2 1
...
car\1#5 2 elevator\1#1 1  lift\1#8 1
car\1#5 22 compartment\1#2 ...  well\1#5 1
```

Figure 4: Format of word contexts

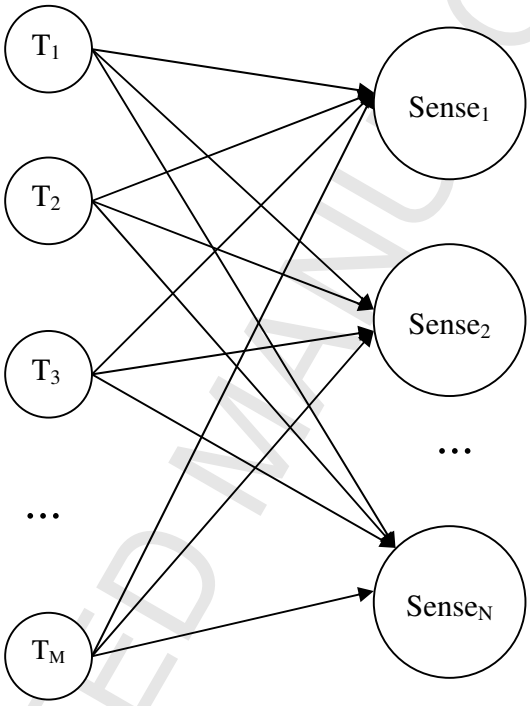


Figure 5: The WSD network architecture for each domain

```
car\1#1 2159 a_couple_of\5#1 0.00891 a_few\5#1 0.04700 a_little\4#1 0.01332 ...
youth\1#1 0.02271 youth\1#3 0.00348 zenith\1#1 0.00815

car\1#2 208 a_few\5#1 -0.00094 afternoon\1#1 0.07513 ... wheelhouse\1#1 -0.01176
whole\1#2 -0.03796 whole_thing\1#1 -0.03796 wink\2#2 0.10061 yard\1#3 0.09999
...
car\1#5 47 area\1#4 0.01380 artefact\1#1 -0.00919 ... wheeled_vehicle\1#1 -0.02466
wheelhouse\1#1 -0.02076 whole\1#2 -0.00919 whole_thing\1#1 -0.00919
```

Figure 6: Format of the codebook vectors after training

Table 1: Contingency Table for i category

	YES is correct	NO is correct
YES is assigned	a_i	b_i
NO is assigned	c_i	d_i

Table 2: Results obtained with the ModApte partition

	P_{macro}	P_{μ}
Rocchio	0.51	0.59
LVQ	0.61	0.73

Table 3: Contingency table for 'earn' category

	YES is correct	NO is correct
YES is assigned	1,047	256
NO is assigned	36	1,521

Table 4: Contingency table for 'coffee' category

	YES is correct	NO is correct
YES is assigned	21	185
NO is assigned	1	2,653

Table 5: Official results for English Lexical Sample

		Fine-grained		Coarse-grained	
	System	Precision	Recall	Precision	Recall
1	HTSA3	0.729	0.729	0.793	0.793
2	ITC-IRST	0.726	0.726	0.795	0.795
...	...	...	...	...	
32	LVQ-UJAEN	0.613	0.613	0.695	0.695
...	...	...	...	...	
46	NRC-Coarse2	0.484	0.484	0.757	0.757
47	DLSI-UA-LS-SU	0.782	0.310	0.828	0.329

Table 6: Official results for English All Words

	System	Precision	Recall
1	GAMBL-AW-S	0.652	0.652
2	SenseLearned-S	0.646	0.646
...	...	...	
9	LVQ-UJAEN	0.590	0.590
...	...	...	
25	autoPSNVs-U	0.359	0.359
26	DLSI-UA-all-Nosu	0.280	0.280