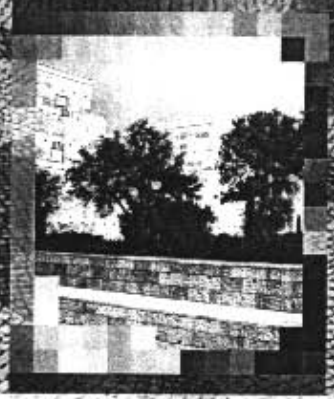
LREC 2004
IV INTERNATIONAL CONFERENCE

EUROPEAN LANGUAGE RESOURCES ASSOCIATION ELRA

In memory of **Antonio Zampolli**

On Language Resources and Evaluation



Lisboa | Portugal

> Centro Cultural de Belém

> 25th May | **Workshops**

- **MEMURA 2004 - Methodologies and Evaluation of Multiword Unit in Real-world Applications**

ELRA
European Language Resources Association

Fourth International Conference on Language Resources and Evaluation

Pre-conference workshop

Workshop Title:

MEMURA 2004 - Methodologies and Evaluation of Multiword Unit in Real-world Applications

Editor:

Gaël Dias, José Gabriel Pereira Lopes, Spela Vintar

Distributed by:

ELRA - European Language Resources Association
55-57 rue Brillat Savarin
75013 Paris
France
Tel.: +33 1 43 13 33 33 / Fax: +33 1 43 13 33 30
http://www.elda.fr
http://www.elra.info

Front cover designed by Orquidea Manuelito

Table of Contents

Japanese Multiword Extraction using SVM and Adaptation 8
T. Ogata, K. Terao and K. Umemura

Multiword Expressions Recognition with the LVQ Algorithm 12
M.C. Díaz-Galiano, M.T. Martín-Valdivia, F. Martínez-Santiago and L.A. Ureña-López

A Parallel Multikey Quicksort Algorithm for Mining Multiword Units 17
R. Pereira, P.Crocker and G. Dias

Recognition and Paraphrasing of Periphrastic and Overlapping Verb Phrases 24
N. Kaji and S. Kurohashi

Transducing Text to Multiword Units 31
C.H.A. Koster

Multiword Units in Syntactic Parsing 39
J. Nivre and J. Nilsson

Use of Noun Phrases in Interactive Search Refinement 47
O. Vechtomova and M. Karamuftuoglu

Comparative Evaluation of C-value in the Treatment of Nested Terms 54
Š. Vintar

Multiword Expressions Recognition with the LVQ Algorithm

M.C. Díaz-Galiano, M.T. Martín-Valdivia, F. Martínez-Santiago, L.A. Ureña-López

Departamento de Informática. University of Jaén.
E-23071. Spain
{mcdiaz, maite, dofer, laurena}@ujaen.es

Abstract

This paper proposes a new neural method based on the supervised Kohonen model for multiword expressions recognition. We use the Learning Vector Quantization algorithm to integrate several statistical estimators to solve this task. Lists of multiword expressions and non-multiword expressions have been generated using the WordNet lexical database to train and test the neural network. Then the neural net has been applied to recognise multiword expressions in a monolingual corpus to prove the effectives in an information retrieval system. The results show that the proposed method is an effective alternative to multiword expressions recognition task.

Introduction

In recent years, there has been growing interest in the Multiword Expressions (MWEs) recognition problem. MWEs are formed by various terms that usually express ideas and concepts that cannot be compressed into a single word. MWEs recognition is very important in many NLP tasks (e.g. machine translation, question-answering, summarisation, etc.). Most real-world applications tend to ignore MWEs or address them simply by listing. However, it is clear that successful applications will need to be able to identify and treat them appropriately.

Methods for automated MWEs recognition have traditionally been statistical (Hull, 1996), (Ballesteros, 1998), and based on the co-occurrence of each particular pair of words in the corpus. Other works (Adriani, 1999) obtain the degree of similarity between terms using the co-occurrence factor, and the standard *tf-idf* term weighting formula. Recently, hybrid approaches incorporating linguistic information have been developed: Diana Maynard and Sophia Ananiadou (Maynard, 2000) make use of different types of contextual information: syntactic, semantic, terminological and statistical. However, different types of information must be managed by integrating them in a given way. The most straightforward is by using a linear function, although this may not be the best way to tackle the problem.

We propose a well-known supervised neural network: Kohonen's Learning Vector Quantization (LVQ) widely used for classification tasks (Kohonen, 1995). The LVQ algorithm will be used to integrate several statistical estimators in order to recognise MWEs.

The rest of the paper is organized as follows. Firstly, we present an introduction to the state of the art, briefly showing some of the currently available methods used to MWEs recognition. These methods include different estimators that will be lately used in our approach. Then we describe the neural network architecture used and the LVQ algorithm. Next section shows the experiments carried out and the results obtained. Finally, we present some conclusions and lines of future work.

Statistical Estimators

Most works what attempt to solve the MWEs recognition problem use estimators to classify terms group. We have integrated several of these estimators, which obtain good results separately, in a neuronal network, trying select a set of heterogeneous and representative estimators.

We have used the following estimators to train and test the neural net:

1. *Pearson's* χ^2. A variant of the χ^2 statistic (Hull, 1996).
2. The *mutual information ratio*, or association ratio, μ (Johansson, 1996).

$$\mu = \log_2\left(\frac{P_{xy}}{P_x \cdot P_y}\right)$$

where P_i is the ocurrence probability of term i in the corpus. This probability is calculated as:

$$P_i = \frac{F_i}{T}$$

where:
F_i = frecuency occurrence of term i.
T = total number of term in the corpus.
Therefore, the first formula can write as:

$$\mu = \log_2\left(\frac{T \cdot F_{xy}}{F_x \cdot F_y}\right)$$

3. Measure the importance of co-occurrence of the elements in a set by the *em* metric (Ballesteros, 1998).

$$em_{xy} = \max\left(\frac{F_{xy} - En(x,y)}{F_x + F_y}, 0\right)$$

where:

$$En(x,y) = \frac{F_x \cdot F_y}{T}$$

where T is the total number of terms in the corpus.

4. *Dice similarity coefficient* obtain the degree of similarity or association-relation between terms using a term association measure and the tf.idf weighting formula (Adriani, 1999).

$$Dice_{xy} = 2\frac{\sum_{i=1}(w'_{xi} \cdot w'_{yi})}{\sum_{i=1} w_{xi}^2 + \sum_{i=1} w_{yi}^2}$$

where:
w_{xi} = the weight of term x in the document i.
w_{yi} = the weight of term y in document i.
$w'_{xi} = w_{xi}$ if term y also occurs in document i, or 0 otherwise.
$w'_{yi} = w_{yi}$ if term x also occurs in document i, or 0 otherwise.
n = the number of documents in the collection.

5. *Dice similarity coefficient*. A variant of the *Dice* estimator (Martinez, 2002).

$$xy = 2\frac{\sum_{i=1}(w'_{xi} \cdot w'_{yi})}{\min\left(\sum_{i=1} w_{xi}^2, \sum_{i=1} w_{yi}^2\right)}$$

Neural Network Architecture

A Neural Network (NN) is a statistical information model that uses learning to adjust the model. NN has been successfully applied in many NLP tasks. In this paper, we propose the use of a competitive neural learning model based on the supervised Kohonen model (Kohonen, 1995) to accomplish the MWEs recognition task: the LVQ algorithm.

MWEs detection is a categorization problem in which only two categories have to be managed: MWE and non-MWE. In our experiment only multiwords with two relevant terms have been used. Consequently, classifying a pair of terms turns into a two step process: firstly, obtain the values yielded by the different estimators; secondly, use those values as inputs for the neural network, and obtain the class to which the pair of terms belongs. More precisely, we have used 5 estimators: *Pearson's* χ^2 (Hull, 1996), the *em* metric (Ballesteros, 1998), the *Dice similarity coefficient* (Adriani, 1999), the *mutual information ratio*, μ (Johansson, 1996) and the *Simpson* coefficient. Figure 1 shows the network architecture.

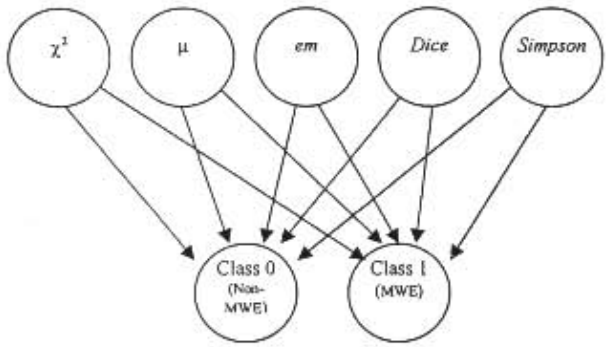


Figure 1: Neural network architecture for MWEs recognition task

In order to train and test our neural network approach, a set of patterns composed of input-output pairs had to be generated, every pattern corresponding to a pair of terms. Thus, to carry out the experiments, we have generated two lists of samples: One list contains vectors with values for the considered estimators for pairs of terms that are MWEs and second list contains vectors with values for the considered estimators for pairs of terms that are non-MWEs. Each vector is labelled with the class to which it belongs (class 1 for the vectors that belong to the MWEs class and class 0 for the vectors that belong to the non-MWEs class).

We have used the CLEF[1] 2001 collection data in order to generate both samples lists. The values for MWEs were obtained by applying the considered estimators to pairs of terms present in the corpus that are MWEs, and labelling the output with the class 1. We consider that a pair of terms is a MWEs if it appears in WordNet (Miller, 1995). WordNet is a lexical database where MWEs can be found. However, not all the pairs of terms said to be MWEs really were. For this reason, each MWE returned by WordNet was checked against in the machine readable dictionary Encarta[2] to remove pairs of terms which were not real MWEs, even though they appeared together very frequently.

A Non-MWE list was also taken from the corpus used in CLEF 2000. Pairs of terms were taken from this corpus and then searched in WordNet, checking that they did not appear in it. If they did not appear, they were once more

[1] The Cross Language Evaluation Forum (CLEF) is an annual activity of European ambit, held since 2000 and coordinated by DELOS Network of Excellence for Digital Libraries conferences, in collaboration with the NIST and the TREC. CLEF aims to promote research and development in CLIR. For more information, see: http://www.clef-campaign.org.

[2] Encarta is a machine readable dictionary available at http://www.encarta.com.

checked against the Encarta dictionary to assure they were not MWEs. The output values for non- MWEs were labelled with the 0 class.

The LVQ Algorithm

To train and test the neural network we have used the supervised Kohonen model: the LVQ algorithm (Kohonen, 1995). This learning algorithm is a classification method based on neural competitive learning, which permits the definition of a group of categories in the space of input data by a reinforced learning. LVQ uses supervised learning to define class regions in the input data space. To this end, a subset of similarly labelled codebook vectors is placed into each class region.

Given a sequence of input data, an initial group of reference vectors w_k (codebook) is selected. In each iteration, a input vector x_i is selected and the vectors W are updated, so that they fit x_i in a better way. The LVQ algorithm works as follows:

For each class, k, a weight vector w_k is associated. In each repetition, the algorithm selects an input vector, x_i , and compares it with every weight vector, w_k, using the euclidean distance $\|x_i-w_k\|$, so that the winner will be the weight vector w_c nearest to x_i, being c its index:

$$\left\|x_i - w_c\right\| = \min_k \left\{\left\|x_i - w_k\right\|\right\}$$

i.e.,

$$c = \arg\min_k \left\|x_i - w_k\right\|$$

The classes compete between themselves in order to find the most similar to the input vector, so that the winner is the one with shorter Euclidean distance with regard to the input vector. Only the winner class will modify its weights using a reinforced learning algorithm, either positive or negative, depending on the classification being correct or not. Thus, if the winner class and the input vector have the same class (the classification has been correct), it will increase the weights, coming slightly closer to the input vector. On the contrary, if the winner class is different from the input vector class (the classification has not been correct), it will decrease the weights, moving slightly further from the input vector.

Let $x_i(t)$ be an input vector at time t, and $w_k(t)$ represents the weight vector for the class k at time t. The following equation defines the basic learning process for the LVQ algorithm.

$$w_c(t+1) = w_c(t) + s \cdot \alpha(t) \cdot \left[x_i(t) - w_c(t)\right]$$

where $s = 0$, if $k \neq c$; $s = 1$, if $x_i(t)$ and $w_c(t)$ belong to the same class; and $s = -1$, if they do not, and where $\alpha(t)$ is the learning rate, being $0<\alpha(t)<1$, a monotonically decreasing function of time. In our experiments we have used $\alpha(t) = 0.3$.

Once the training phase has finished, the production phase starts. Again, each testing vector is presented to the network input. The original LVQ algorithm must find the winner class calculating the Euclidean distances between the codebook vectors and input vectors. The winner class will be the codebook vectors with the shortest distance with regard the input vector.

In order to improve the network precision, we have used a modified version of the LVQ algorithm during the evaluation phase. An input vector is presented to the neural network but the output network is the distance to the codebook vector belonging to the MWEs class (class labelled with 1). This value represents a confidence score assigned by the neural network to the pair of terms considered is a MWE. Thus, during the evaluation the non-MWE class is no considered. The network output is normalized and finally is inverted by subtracting 1. Thus, value near to 1 indicates a high confidence in the input vector represents a MWE.

Evaluation and Results

As we have commented previously, the experiments have been carried out for the English CLEF 2001 collection data. The collection is composed by 113,005 news from the *Los Angeles Time* newspaper edition 1994, 50 queries and their relevance assessments.

The corpus has been pre-processed as usual in information retrieval systems (Frakes and Baeza-Yates, 1992), using stopword lists and stemming algorithms available via the Web[3]. Stopword lists have been increased with terms such as "retrieval", "documents", "relevant", etc.

Next step consists of generating the two lists of samples from corpus. Once both MWEs and non-MWEs lists had been created, the estimators were applied to them, obtaining the file with the patterns to be used with the supervised network. This file was split to use 50% of the patterns to train the neural network and the remaining 50% to test it. A total of 1,000 training vectors and 1,000 test vectors were generated.

To test the neural network we have modified the LVQ algorithm. The original LVQ must find the winner class by calculating the Euclidean distances between the codebook vectors and input vectors. The winner class will be the codebook vectors with the shortest distance with regard to the input vector.

The experiments were carried out using the implementation described in LVQ_PAK documentation (Kohonen, 1991) with default parameters. Thus, every experiment used two codebook vectors (one per class) and the learning rate started at 0.3.

The experiments were carried out with the original LVQ and with the modified proposed version for 4 confidence values (0.95, 0.90, 0.80, 0.70). The original LVQ network recognises correctly 684 MWE over 1000 test vectors. Therefore, the original LVQ obtain a 68.40% of precision. The results of the LVQ modified are more hopeful. Table 1 shows the precision obtained when we consider only the test vectors for which the network output overcomes the confidence values. With a confidence value of 0.95 the precision obtained is a 100%. In proportion to the threshold is smaller, the precision also decrease. For 0.90 of threshold the precision is 94.84%, only 74 good MWEs are recognised. With confidences values of 0.80 and 0.70 the precisions are 88.59% and 81.10% respectively. The precision obtained increase when the confidence value increase.

[3] http://www.unine.ch/info/clef

	Patterns considered	Successfully MWE detected
LVQ-0.95	40	40
LVQ-0.90	78	74
LVQ-0.80	184	163
LVQ-0.70	291	236

Table 1: Precision obtained with several confidence values

The results obtained are very good separately, which demonstrates that LVQ network works good in the MWEs recognition task with the sample data available. In order to prove the effectives in an information retrieval system, we have indexed the CLEF 2001 English collection.
Once the collection has been pre-processed and the MWEs have been recognised, we indexed the corpus with the Zprise information retrieval system[4], using the OKAPI probabilistic model (Robertson, Walker and Beaulieu, 2000). We have generated 6 different indexes

- Index without MWEs (baseline). We use the word as indexation unit.
- Index with MWEs recognised by the original LVQ network (binary output). We use words and MWEs recognised by the neural network as indexation unit.
- Index with MWEs recognised by the modified LVQ evaluation algorithm with a confidence value of 0.95. The indexation units are words and MWEs recognised by the neural network using the modified LVQ algorithm. In this case, MWEs are considered only those that overcome the confidence value of 0.95.
- Index with MWEs recognised by the modified LVQ evaluation algorithm with a confidence value of 0.90. The indexation units are words and MWEs recognised by the neural network using the modified LVQ algorithm. In this case, MWEs are considered only those that overcome the confidence value of 0.90.
- Index with MWEs recognised by the modified LVQ evaluation algorithm with a confidence value of 0.80. The indexation units are words and MWEs recognised by the neural network using the modified LVQ algorithm. In this case, MWEs are considered only those that overcome the confidence value of 0.80.
- Index with MWEs recognised by the modified LVQ evaluation algorithm with a confidence value of 0.70. The indexation units are words and MWEs recognised by the neural network using the modified LVQ algorithm. In this case, MWEs are considered only those that overcome the confidence value of 0.70.

The experiments have been carried out by using about 105.000 news published in Los Angeles Times for 1994. In order to evaluate such experiments, we have used 50 queries and their relevance assessments from CLEF 2001 workshop. Only Title and Description sections have been taking into account. Finally, we have built six indices according to the patterns considered (without MWEs, LVQ, LVQ-0.95, LVQ-0.90, LVQ-0.80 and LVQ-070). Table 2 shows the precision obtained with the 6 indexes considered.

	Precision
Without MWEs	0.458
LVQ	0.424
LVQ-0.95	0.509
LVQ-0.90	0.471
LVQ-0.80	0.461
LVQ-0.70	0.427

Table 2. Precision obtained with MWEs recognition in an information retrieval system

The obtained results aim MWEs are useful in the information retrieval task only when precision is very high. The reason: whether erroneous MWEs are labelled then the precision achieved by the information retrieval system precision falls off dramatically. Thus, high precision is preferable even some real multiword expressions are not taking into account by the information retrieval system.

Conclusions and Future Works

We have proposed a new neural approach to MWEs recognition. The neural network uses the Kohonen LVQ algorithm. To train and test the network we have generated two lists of sample of MWEs and non-MWEs. When we evaluate the neural networks separately, the results obtained are very promising. But it does not happen the same way when the network is applied to MWEs recognition in an information retrieval system since the performance is seriously damaged by erroneous MWEs.
We could apply the proposed MWEs recognition method to improve precision in other natural language processing tasks such as summarization system, machine translation or question-answering. These tasks need a high precision in the MWEs recognition.

Acknowledgements

This work has been supported by Spanish Government (MCYT) with grant TIC2003-07158-C04-04.

References

(Adriani, 1999) M. Adriani, C.J. van Rijsbergen. Term Similarity Based Query Expansion for Cross Language Information Retrieval. *Proceedings of Research and Advanced Technology for Digital Libraries*, 311-322, 1999.

(Ballesteros, 1998) L. Ballesteros, W.B. Croft. Resolving ambiguity for cross-language retrieval. *Proceedings of the 21st Annual International ACM SIGIR Conference*, 64-71, 1998.

(Frakes and Baeza-Yates, 1992) W.B. Frakes, R. Baeza-Yates. Information retrieval: Data, structures and algorithms. Pretice Hall, 1992.

[4] Zprise is an information retrieval system developed by Darrin Dimmick (NIST). Available on demand at http://www.itl.nist.gov/iaui/894.02/works/papers/zp2/zp2.html

(Hull, 1996) D.A. Hull, G. Grefenstette. Experiments in Multilingual Information Retrieval. *Proceedings of the 19th Annual International ACM SIGIR Conference*, 1996

(Johansson, 1996). C. Johansson. Good Bigrams. *Proceedings COLING-96*. 592-597, 1996

(Kohonen, 1995) T. Kohonen, Self-Organization and Associative Memory. 2nd Ed. Springer-Verlag, Berlin, 1995.

(Kohonen, 1996) T. Kohonen, J. Hynninen,J. Kangas, J. Laaksonen, K. Torkkola. LVQ_PAK: The Learning Vector Quantization program package. *Technical Report A30*, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 1996.

(Maynard, 2000) D. Maynard, S. Ananiadou. TRUCKS: a model for automatic term recognition, *Journal of Natural Language Processing*, 2000.

(Miller, 1995) G. Miller. WORDNET: A lexical database for English. *Communications of the ACM*, 38 (11), 1995.

(Martínez, 2002) F. Martínez, M.T. Martín, V.M. Rivas, M.C. Díaz, L.A. Ureña. Using Neural Networks for Multiword Recognition in IR. *Proceedings of the 7th Internacional ISKO Conference*, 2002.

(Robertson, Walker and Beaulieu, 2000) Robertson, S.E., S. Walker y M. Beaulieu. 2000. Experimentation as a way of life: okapi at TREC. *Information Processing and Management*. Vol, 1, pp. 95-108