Honkela, T., Hyvärinen, A. and Väyrynen, J.J. (2003). Emergence of Linguistic Representations by Independent Component Analysis. Helsinki University of Technology, Publications in Computer and Information Science, Report A72.

Emergence of Linguistic Representations by Independent Component Analysis

Timo Honkela[1], **Aapo Hyvärinen**[2] and Jaakko J. Väyrynen[1]

[1]Helsinki University of Technology
Neural Networks Research Center
Laboratory of Computer and Information Science
timo.honkela@hut.fi

[2]Helsinki Institute of Information Technology
Basic Research Unit
Department of Computer Science
University of Helsinki
aapo.hyvarinen@helsinki.fi

December 15, 2003

Abstract

Our aim is to find syntactic and semantic relationships and roles of words based on the analysis of corpora. We study three methods for analyzing words in contexts as potential methods for solving this task. The methods are latent semantic analysis, self-organizing map and independent component analysis. Latent semantic analysis is a simple method for automatic generation of concepts that are useful, e.g., in encoding documents for information retrieval purposes. However, these concepts cannot easily be interpreted by humans. Self-organizing maps can be used to generate an explicit diagram which characterizes the relationships between

words. A word map reflects syntactic categories in the overall organization and semantic categories in the local level. The self-organizing map does not, however, provide any explicit distinct categories for the words. The emergent syntactic and semantic categories are only implicit. Independent component analysis applied on word context data gives distinct features which reflect syntactic and semantic categories. Thus, independent component analysis gives features or categories that are both explicit and can easily be interpreted by humans. This result can be obtained without any human supervision or tagged corpora that would have some predetermined morphological, syntactic or semantic information.

1 Introduction

A word can belong to several syntactic categories simultaneously. The number of categories is even higher if one takes into account the semantic categories. Traditionally, such categorization is determined by hand: the categories into which a word belongs to are described in a dictionary.

In the following, we will study the emergence of linguistic representations through the analysis of words in contexts. First, we give a general description of the approach and describe two methods that have widely been used for the analysis, latent semantic analysis and self-organizing map. Then we introduce a novel approach based on independent component analysis.

1.1 Analysis of Words in Contexts

Contextual information has widely been used in statistical analysis of natural language corpora (consider, e.g., (Church and Hanks, 1990; Schütze, 1992; Manning and Schütze, 1999)). Handling computerized form of written language rests on processing of discrete symbols. How can a symbolic input such as a word be given to a numeric algorithm? Similarity in the appearance of the words does not usually correlate with the content they refer to. As a simple example one may consider the words “window”, “glass”, and “widow”. The words “window” and “widow” are phonetically close to each other, whereas the semantic relatedness of the words “window” and “glass” is not reflected by any simple metric.

One useful numerical representation can be obtained by taking into account the sentential context in which the words occur. First, we represent each word by a vector in an n-dimensional space, and then code each context as an average of vectors representing the words in that context. In the simplest case, the dimension n can be taken equal to the number of different words, and each word is represented by a vector with one element equal to one and others equal to zero. Then the context vector simply gives the frequency of each word in the context. For computational reasons, however, the dimension may be reduced by different methods. In information retrieval, a similar approach is called bag-of-words applied in methods, e.g., related to the vector space model (Salton et al., 1975).

1.2 Latent Semantic Analysis

In latent semantic analysis (Deerwester et al., 1990), a technique known as singular value decomposition (SVD) is used to create a latent semantic space. First, a term by document matrix **A** is generated. Every term is represented by a row in matrix **A**, and every document is represented by a column. An individual entry in **A**, a_{ij}, represents the frequency of the term i in document j. Next, SVD is used to decompose matrix **A** into three separate matrices. The first matrix is a term by concept matrix **B**. The second matrix is a concept by concept matrix **C**. The third matrix is a concept by document matrix **D**. The context for each word in the basic LSA is the whole document. This is a special case of the coding of contexts explained in above: the context is one whole document in the LSA.

In (Landauer and Dumais, 1997) the LSA is described in terms of learning and cognitive science. The claim is that the LSA acquired knowledge about the full vocabulary of English at a comparable rate to school-children. The development of the LSA has also been motivated by practical applications (Furnas et al., 1987).

One problem with the LSA is that the concept space is difficult to understand by humans. The self-organizing map, that will be introduced in the next section, creates a visual display of the analysis results which is readily understandable for a human viewer.

1.3 Self-Organizing Map of Words

The self-organizing map (Kohonen, 1982) can be visualized as a two-dimensional, sheet-like grid. The grid consists of a number of processing elements (units or nodes). In a self-organizing map, the nodes become specifically tuned to various inputs in an orderly fashion. The learning process is unsupervised. A much more detailed description of the method and its numerous applications can be found in (Kohonen, 2001).

Earlier, the self-organizing map has been used in the analysis of word context data, e.g., by (Ritter and Kohonen, 1989) (artificially generated short sentences), and (Honkela et al., 1995) (Grimm fairy tales). In (Finch and Chater, 1992), a self-organizing map analysis of word contexts was performed with a one-dimensional map in order to find synonymous words. The results of an analysis similar to the one conducted in (Honkela et al., 1995) are shown in Fig.12. The result can be called a self-organizing map of words, or a word category map. Earlier, the name self-organizing semantic map has also been used. Similar results have also been presented by Miikkulainen (Miikkulainen, 1990; Miikkulainen, 1993; Miikkulainen, 1997). The analysis here is made for illustration purposes and one should consider (Ritter and Kohonen, 1989) and (Honkela et al., 1995) for more thorough analysis and explanation of the methodology.

The map was made using SOM Toolbox for Matlab (Vesanto et al., 2000). The analysis used the *som_make* function that creates, initializes, and trains a SOM using default parameters with map size of 120 units. The data was a collection of e-mails sent to the connectionists list. One hundred common words were chosen to be mapped and the contextual information was calculated using the 2000 most common types. The preparation of the data will be discussed in more detail in Chapter 2.1.

Areas or local regions on a word category map can be considered as implicit categories or classes that have emerged during the learning process. Single nodes in the map can be considered as adaptive prototypes. Each prototype is involved in the adaptation process in which the neighbors influence each other and the map is gradually finding a form in which it can best represent the input.

One classical approach for defining concepts is based on the idea that a concept can be characterized by a set of defining attributes. In contrast, the prototype theory of concepts involves that concepts have a prototype structure and there is no delimiting set of necessary and sufficient conditions for determining category membership that can also be fuzzy. Instances of a concept can be ranked in terms of their typicality. Membership in a category is determined by the similarity of an object's attributes to the category's prototype. The development of prototype theory is based on the works by, e.g., Rosch (Rosch, 1977) and Lakoff (Lakoff, 1987).

MacWhinney (MacWhinney, 1989) discusses the merits and problems of the prototype theory. He mentions that prototype theory fails to place sufficient emphasis on the relations between concepts. MacWhinney also points out that prototype theory has not covered the issue of how concepts develop over time in language acquisition and language change, and, moreover, it does not provide a theory of representation. MacWhinney's competition model has been designed to overcome these deficits. MacWhinney has presented a model of emergence in language based on the SOM (MacWhinney, 1997). Recently, (Gärdenfors, 2000) has presented theoretical foundations for conceptual modeling in which the SOM is an important element (consider also (Gärdenfors, 1996; Gärdenfors, 1997)).

The emergent categories on a word category map are implicit. The borderlines for any categories have to be determined separately. It would be beneficial if one could find the categories in an automated analysis. Moreover, each word appears in one location of the map. This means, among other things, that one cannot have a map in which several characteristics or categories of one word would be represented unless the the categories overlap and accordingly the corresponding areas of the map overlap. In some cases, this is the case: it is possible to see the area of modal verbs inside the area of verbs, e.g., in the map in (Honkela et al., 1995). However, one might wish to find a sparse encoding of the words in such a way that there would be a collection of features associated with each word. For instance, a word can be a verb, a copula and in past tense. It is an old idea in linguistics to associate words with features. The features can be syntactic as well as semantic like, e.g., proposed already in (Fillmore, 1968). However, these features are given by hand, and the membership is crisp.

1.4 Independent Component Analysis of Word Contexts

In the following, we propose the use of independent component analysis (ICA) (Comon, 1994; Jutten and Hérault, 1991; Hyvärinen et al., 2001) for the extraction of linguistic features from text corpora and present a detailed methodological description. ICA learns features in an unsupervised manner. Several such features can be present in a word, and ICA gives the explicit values of each feature for each word. We expect the features to coincide with known syntactic and semantic categories: for instance, we expect ICA to be able to find a feature that is shared by words such as “must”, “can” and “may”. In earlier studies, independent component analysis has been used for document level analysis of texts (see, e.g., (Bingham et al., 2002)).

2 Data and methods

2.1 Data collection

The data used in the experiments consists of collection of e-mails sent to the connectionists mailing list[1]. The texts were concatenated into one file. Punctuation marks were removed and all uppercase letters were replaced by the corresponding lowercase letters. The resulting corpus consists of $4,921,934$ tokens (words in the running text) and $117,283$ types (different unique words).

One hundred common words were manually selected and the contextual information was calculated using the 2000 most common types. We formed a context matrix $\mathbf{C}$ in which c_{ij} denotes the number of occurrences jth word in the immediate context of ith word, i.e, ith word followed by jth word with no words between them. This provided a 100×2000 matrix. A logarithm of the number of occurrences was taken in order a reduce the effect of the very most common words in the analysis and finally each word vector was normalized to unit length.

2.2 Independent component analysis

We will give a brief outline of the basic theory of independent component analysis (Hyvärinen et al., 2001). The classic version of the ICA model can be expressed as

$$\mathbf{x} = \mathbf{A}\mathbf{s} \tag{1}$$

[1] http://www-2.cs.cmu.edu/afs/cs.cmu.edu/project/connect/connect-archives/

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ is the vector of observed random variables, the vector of the independent latent variables is denoted by $\mathbf{s} = (s_1, s_2, \ldots, s_n)^T$ (the "independent components"), and $\mathbf{A}$ is an unknown constant matrix, called the mixing matrix. If we denote the columns of matrix $\mathbf{A}$ by a_j the model can be written as

$$\mathbf{x} = \sum_{I=1}^{n} a_i s_i \tag{2}$$

The goal in ICA is to learn the decomposition in Eq. (1) in an unsupervised manner. That is, we only observe $\mathbf{x}$ and want to estimate both $\mathbf{A}$ and $\mathbf{s}$. ICA can be seen as an extension to principal component analysis and factor analysis which underlie LSA. However, ICA is a more powerful technique capable of finding the underlying factors when the classic methods would fail.

The starting point for the ICA is the simple assumption that the s_i are statistically independent. Two variables, y_1 and y_2, are independent if information on the value of y_1 does not give any information on the value of y_2, and vice versa. This does not need to hold for the observed variables x_i. In case of two variables, the independence holds if and only if $p(y_1, y_2) = p(y_1)p(y_2)$. This definition extents to any number of random variables.

The are three properties of the ICA that should be taken into account when considering the analysis results. First, one cannot determine the variances of the independent components s_i. The reason is that, both $\mathbf{s}$ and $\mathbf{A}$ being unknown, any scalar multiplier in one of the sources s_i could always be canceled by dividing the corresponding column a_i of $\mathbf{A}$ by the same scalar. As a normalization step, one can assume that each component has a unit variance, $E\{{s_i}^2\} = 1$. The ambiguity of the sign still remains: one could multiply a component by -1 without affecting the model.

The second property to be remembered is that one cannot determine the order of the components. While both $\mathbf{s}$ and $\mathbf{A}$ are unknown one can freely change the order of the terms in Eq. (2) and call any of the components the first one.

The third important property of ICA is that the independent components must be nongaussian for ICA to be possible (Hyvärinen et al., 2001). Then, the mixing matrix can be estimated up to the indeterminacies of order and sign discussed above. This is in stark contrast to such techniques as principal component analysis and factor analysis, which are only able to estimate the mixing matrix up to a rotation, which is quite insufficient for our purposes.

For our ICA analyses we applied FastICA[2] software package for Matlab. We fed the word-context matrix **C** to the FastICA algorithm (Hyvärinen, 1999) so that each column was considered one data point, and each row one random variable.

We used the standard maximum-likelihood estimation by setting the nonlinearity g to the tanh function, and using symmetric orthogonalization (Hyvärinen et al., 2001) (p. 212). The dimension of the data was reduced to 10 by principal component analysis (this is implemented as part of the software)[3]. Reduction of the dimension is often used to reduce noise and overlearning (Hyvärinen et al., 2001) (p. 267).

3 Results

The results of the ICA analysis corresponded in most cases very well or at least reasonably well with our preliminary intuitions. The system was able to automatically create distributed representations as a meaningful collection of emergent linguistic features; each independent component was one such feature.

In the following, we will show several examples of the analysis results. In considering the feature distributions, it is good to keep in mind that the sign of the features is arbitrary. As was mentioned earlier, this is because of the ambiguity of the sign: one could multiply a component by -1 without affecting the model (see Section 4.1).

Fig. 1 shows how the third component is strong in the case of nouns in singular form. A similar pattern was present in all the nouns with three exceptional cases with an additional strong fourth component indicated in Fig. 2. The reason appears to be that “psychology”, “neuroscience”, and “science” share a semantic feature of being a science or a scientific discipline. This group of words provide a clear example of distributed representation where, in this case, two components are involved.

An interesting point of comparison for Fig. 1 is the collection of plural forms of

[2] $http://www.cis.hut.fi/projects/ica/fastica/$

[3] The Matlab code for the operations was as follows:
$LC = log(C+1);$
$[A,W] = fastica(LC,'approach','symm','g','tanh','lastEig',10,'epsilon',0.0005);$

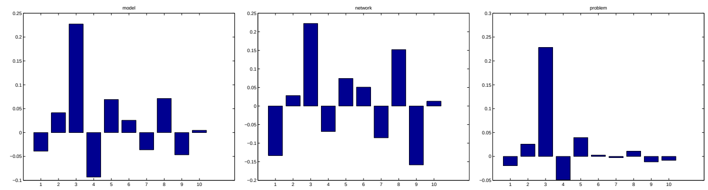

Figure 1: ICA features for “model”, “network” and “problem”.

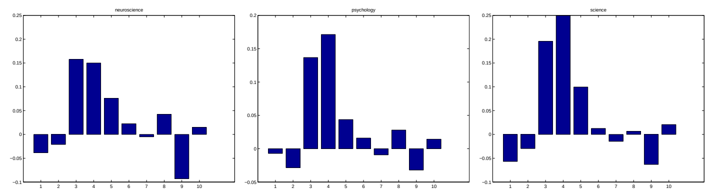

Figure 2: ICA features for “neuroscience”, “psychology” and “science”.

the same nouns in Fig. 3. The third component is strong as with the singular nouns but now there is another strong component, the fifth.

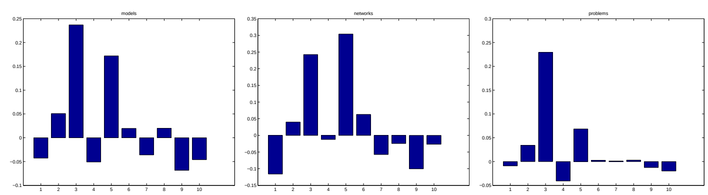

Figure 3: ICA features for “models”, “networks” and “problems”.

Fig. 4 shows how all the possessive pronouns share the feature number nine.

Modal verbs are represented clearly with component number ten as shown in Fig. 5. Here, slightly disappointingly the modal verbs are not directly linked with verbs in general through a shared component. This may be because of the distinct nature of the modal verbs. Moreover, one has to remember that in this analysis we used ten as the number of ICA features which sets a limit on the complexity of the feature encoding. We used this limit in order to demonstrate the powerfulness and usefulness of the method in a simple manner. A higher number of features can be used in order to obtain more detailed feature distinctions.

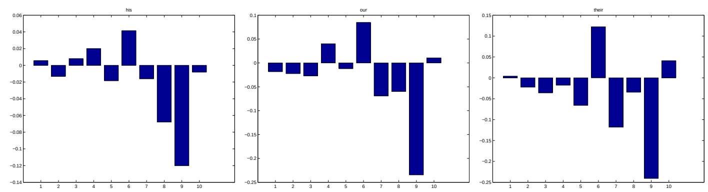

Figure 4: ICA features for “his”, “our” and “their”.

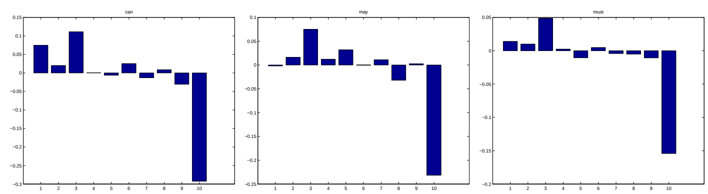

Figure 5: ICA features for “can”, “may” and “must”.

Fig. 6 shows how the adjectives are related to each other through the shared feature number eight, and even number nine in the opposite direction. Quite interestingly this component number nine is associated with ing-ending verbs (see Fig. 7 such as “modeling”, “training”, and “learning” that can, naturally, serve in the position of an adjective or a noun (consider, for instance, “training set” versus “network training”).

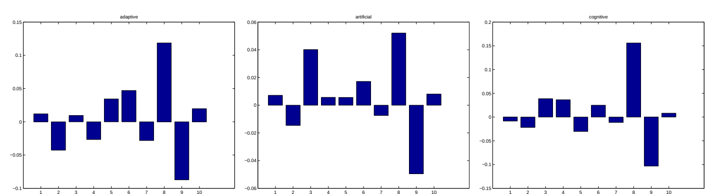

Figure 6: ICA features for “adaptive”, “artificial” and “cognitive”.

Fig. 8 shows how the three articles use two feature dimensions, namely the sixth and seventh.

Finally, there are individual words, particularly some verbs for which the result is not as clear as for other words. In Fig. 9 it is shown how the verb “include” and the copula “is” have several features present in a distributed manner. The

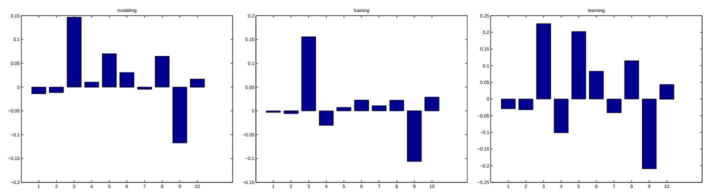

Figure 7: ICA features for “modeling”, “training” and “learning”.

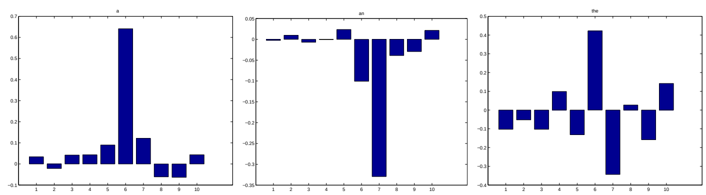

Figure 8: ICA features for “a”, “an” and “the”.

word “is” shares, however, clearly the feature number two with the word “have”. This slight anomaly particularly concerning “include” may also be related to the fact that ten features were used for the hundred words. For a related reason, a collection of particles and similar common words were excluded in the analysis because many of them are rather unique in their use considering the contexts in which their appear. This phenomenon was already discernable in the analysis word contexts using the self-organizing map (Honkela et al., 1995).

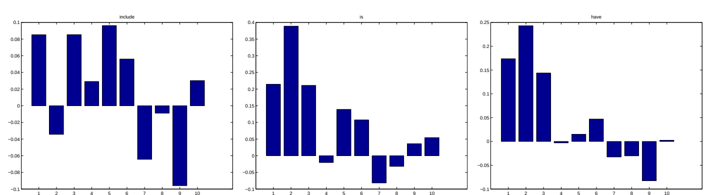

Figure 9: ICA features for “include”, “is” and “have”.

The categorical nature of each component can also be illustrated by listing the words that are strongest in each component (see Fig. 10 and Fig. 11). The result shows some very clear components such as 3 to 5 which can be considered noun categories. These three components were already discussed earlier. Component number 8 is populated by adjectives whereas number 10 contains modal verbs. Verbs “to be” and “have” are in their different forms in the component 2. We can also see a certain kind of component overloading in components 1 and 2. This is explained by the limited number of component in use. With a larger number of components, more detailed categories can be gained and ambiguity inside a category can be avoided.

1	2	3	4	5
or	is	paper	science	networks
and	are	information	university	systems
is	have	it	engineering	learning
are	has	papers	research	models
have	i	system	psychology	processing
has	we	work	neuroscience	algorithms
use	they	networks	technology	recognition
...	...	...	...	...

Figure 10: The most representative words for the first five features (components), in the order of representativeness, top is highest.

The nouns “network” and “control” in component 8 in Fig. 11 are often used in the corpus in noun phrases like “neural network society”. In general, the area and style of the texts in the corpus are, of course, reflected in the analysis results.

6	7	8	9	10
a	the	neural	their	will
the	an	computational	our	can
and	and	cognitive	your	may
or	or	network	my	should
their	their	adaptive	learning	would
its	its	control	research	must
your	are	learning	processing	did
...	...	...	...	...

Figure 11: The most representative words for the last five features (components), in the order of representativeness, top is highest.

4 Discussion

In this article, we started by discussing some advantages and limitations of latent semantic analysis and the self-organizing maps in the analysis of word contexts. Latent semantic analysis suffers from the limitation that the underlying semantic space remains implicit. The self-organizing map is able to explicate the semantic space as relationships on the map. However, the categories remain implicit and there is only one position for each word in the map which is a limitation considering the intuitive idea that a word may very well belong to several categories simultaneously.

We have shown how independent component analysis can bring an additional advantage of finding explicit features that characterize words in an intuitively appealing manner. We have considered the methods for the analysis of words as they appear in text corpora. All these methods are beneficial as automatic statistical methods for linguistic analysis. However, independent component analysis appears to make possible a qualitatively new kind of result which have earlier been obtainable only through hand-made analysis.

The analysis results show how the ICA analysis was able to reveal underlying linguistic features based solely on the contextual information. The results include both an emergence of clear distinctive categories or features as well as a distributed representation based on the fact that a word may belong to several categories simultaneously. For illustration purposes we kept the number of features low, i.e., ten. However, similar approach scales well up to higher numbers of dimensions.

Future research directions include analysis of larger corpora for extracting larger number of independent components. Various options for, e.g., determining the contextual window will be tested. On a qualitative level, polysemy will be considered. Whether the component values can be applied as degrees of membership for each word in each category is a question of further analysis. The distributed representation can be used as a well-motivated low-dimensional encoding for words in various applications. The limited number of dimensions brings computational efficiency whereas the meaningful interpretation of each component provides basis for intelligent processing. To interpret the estimated components as linguistic features, it is necessary to measure how well they capture linguistic information. We will also study the closeness of match between the emergent components and manually determined linguistic categories[4].

We are optimistic that the approach will be relevant in areas like language technology in the form of practical applications in information retrieval and machine translation; as well as in cognitive linguistics as a provider of additional understanding on potential cognitive mechanisms in natural language learning and understanding.

[4]This research is currently conducted by Jaakko J. Väyrynen.

References

Bingham, E., Kuusisto, J., and Lagus, K. (2002). Ica and som in text document analysis. In *Proceedings of the 25th ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 361–362.

Church, K. and Hanks, P. (1990). Word association norms, mutual information and lexicography. *Computational Linguistics*, 16:22–29.

Comon, P. (1994). Independent component analysis—a new concept? *Signal Processing*, 36:287–314.

Deerwester, S., Dumais, S., Landauer, T., Furnas, G., and R.A. Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41:391–407.

Fillmore, C. (1968). *Universals in Linguistic Theory*, chapter The case for case, pages 1–88. Holt, Rinehart and Winston, Inc.

Finch, S. and Chater, N. (1992). Unsupervised methods for finding linguistic categories. In Aleksander, I. and Taylor, J., editors, *Artificial Neural Networks, 2*, pages II–1365–1368. North-Holland.

Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). The vocabulary problem in human-system communication. *Communications of the ACM*, 30(11):964–971.

Gärdenfors, P. (1996). Mental representation, conceptual spaces and metaphors. *Synthese*, 106:21–47.

Gärdenfors, P. (1997). *Mindscapes: Philosophy, Science, and the Mind*, chapter Meanings as conceptual structures, pages 61–86. Pittsburgh University Press, Pittsburgh.

Gärdenfors, P. (2000). *Conceptual Spaces*. MIT Press.

Honkela, T., Pulkki, V., and Kohonen, T. (1995). Contextual relations of words in Grimm tales analyzed by self-organizing map. In *Proceedings of ICANN-95, International Conference on Artificial Neural Networks*, volume 2, pages 3–7. EC2 et Cie.

Hyvärinen, A. (1999). Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks*, 10:626–634.

Hyvärinen, A., Karhunen, J., and Oja, E. (2001). *Independent Component Analysis*. John Wiley & Sons.

Jutten, C. and Hérault, J. (1991). Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. *Signal Processing*, 24:1–10.

Kohonen, T. (1982). Self-organizing formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kohonen, T. (2001). *Self-Organizing Maps*. Springer, Berlin, Heidelberg. as a book.

Lakoff, G. (1987). *Women, Fire and Dangerous Things*. University of Chicago Press, Chicago.

Landauer, T. and Dumais, S. (1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction and representation of knowledge. *Psychological Review*, 104:211–240.

MacWhinney, B. (1989). *Linguistic categorization*, chapter Competition and Lexical Categorization. Benjamins, New York.

MacWhinney, B. (1997). *Cognitive approaches to language learning*, chapter Lexical Connectionism. MIT Press.

Manning, C. and Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press, Cambridge, MA.

Miikkulainen, R. (1990). A distributed feature map model of the lexicon. In *Proceedings of 12th Annual Conference of the Cognitive Science Society*, pages 447–454, Hillsdale, NJ. Lawrence Erlbaum.

Miikkulainen, R. (1993). *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory*. MIT Press, Cambridge, MA.

Miikkulainen, R. (1997). Self-organizing feature map model of the lexicon. *Brain and Language*, 59:334–366.

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

Rosch, E. (1977). *Studies in cross-cultural psychology, vol. 1*, chapter Human categorization, pages 3–49. Academic Press, New York.

Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620.

Schütze, H. (1992). Dimensions of meaning. In *Proceedings of Supercomputing*, pages 787–796.

Vesanto, J., Himberg, J., Alhoniemi, E., and Parhankangas, J. (2000). Som toolbox for matlab 5. Technical Report A57, Helsinki University of Technology, Laboratory of Computer and Information Science, Espoo, Finland.

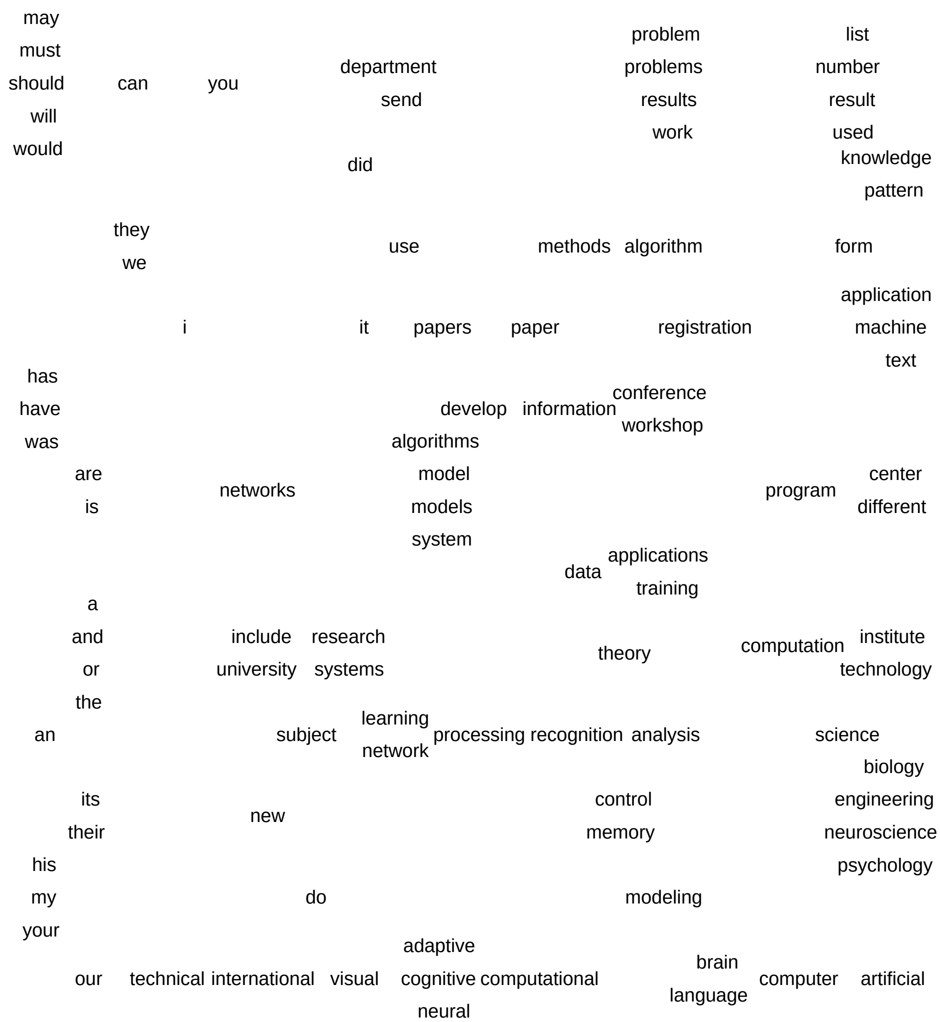

Figure 12: Word category map where the position of each word is determined by the self-organizing map algorithm. The input data for the algorithm consisted of averaged contexts for each word, capturing the information on what are the neighboring words in the text collection.