Appeared in: Proceedings of International Conference on Artificial Neural Networks, ICANN-95, F. Fogelman-Soulie and P. Gallinari (eds.), EC2 et Cie, Paris, 1995, pp. 3-7.

Contextual Relations of Words in Grimm Tales, Analyzed by Self-Organizing Map

Timo Honkela, Ville Pulkki and Teuvo Kohonen

Helsinki University of Technology
Neural Networks Research Centre
Rakentajanaukio 2 C, FIN-02150, Espoo, FINLAND
tel: +358 0 451 3276, fax: +358 0 451 3277
email: Timo.Honkela@hut.fi

Abstract

Semantic roles of words in natural languages are reflected by the contexts in which they occur. These roles can explicitly be visualized by the Self-Organizing Map (SOM). In the experiments reported in this work the source data consisted of the raw text of Grimm fairy tales without any prior syntactic or semantic categorization of the words. The algorithm was able to create diagrams that seem to comply reasonably well with the traditional syntactical categorizations and human intuition about the semantics of the words.

1 Processing Natural Language with Self-Organizing Map

It has earlier been shown that the Self-Organizing Map (SOM) can be applied to the visualization of contextual roles of words, i.e., similarities in their usage in short contexts formed of adjacent words [4]. This paper demonstrates that such relations or roles are also statistically reflected in unrestricted, even quaint natural expressions. The source material chosen for this experiment consisted of 200 Grimm tales (English translation).

In most practical applications of the SOM, the input to the map algorithm is derived from some measurements, usually after their preprocessing. In such cases, the input vectors are supposed to have metric relations. Interpretation of languages, on the contrary, must be based on the processing of sequences of discrete symbols. If the words were encoded numerically, the ordered sets formed of them could also be compared mutually as well as with reference expressions. However, as no numerical value of the code should imply any order to the words themselves, it will be necessary to use uncorrelated vectors for encoding. The simplest method to introduce uncorrelated codes is to assign a unit vector for each word. When all different words in the input material are listed, a code vector can be defined to have as many components as there are words in the list. This method, however, is only practicable in very small experiments. If the vocabulary is large as in the present experiments, we may then encode the words by quasi-orthogonal random vectors of a much smaller dimensionality [4].

To create a map of discrete symbols that occur within the sentences, each symbol must be presented in the due context. The context may consist of the immediate surroundings of the word in the text.

Application of the self-organizing maps to natural language processing has been described earlier in, e.g., [2], [3], [4], [5], and [6].

2 Experiments

2.1 Source data

In the present experiments the data consisted of a set of English translations of fairy tales collected by the Grimm brothers. The number of words in the text was almost 250 000 in total and the size of the vocabulary was over 7000 words. Although the subject area of the tales is rather restricted, the language can be considered to be arbitrarily chosen, and unrestricted. First, the language itself is not formal by any means. This choice of source data is a significant generalization compared with the artificially produced sentences earlier used in simple experiments (e.g. in [4]). Second, the contents of the texts are very diverse.

2.2 Preprocessing

The texts of all tales were concatenated into one file. Punctuation marks were removed and all uppercase letters were replaced by the corresponding lowercase letters. The articles ("a", "an", "the") were also removed from the text. Some tests were made with the articles involved, whereby nouns and personal pronouns became separated to a greater extent.

Word triplets ("predecessor", "key", "successor") picked up from the text file were chosen for the input vector $x(t)$. The triplets were formed simply by taking the encoded representations of three subsequent words from the preprocessed text. All word triplets from the text were collected and stored as a source file.

The 150 most frequent words in the text file were chosen for the "key" words to be represented in the map. It has to be noted, however, that no words were ignored from the original text. There was a coding for each word in the vocabulary, and the predecessor and the successor of the key could be any word occurring in the text.

Encoding of the words was made using a 90-dimensional random real vector for each word. The codes were statistically independent so that there was no correlation between them. The code vectors of the words in the triplet were then concatenated into a single input vector $x(t)$, the dimensionality of which was thus 270.

2.3 Learning Process

The 270-dimensional input vectors $x(t)$ were used as inputs to the SOM algorithm. The SOM array itself was a planar, hexagonal lattice of 42 by 36 formal neurons. For to speed up computations, its codebook vectors were given ordered initial values, chosen from the signal space in the following way. First, a two-dimensional subspace, spanned by the two principal eigenvectors of the input data vectors, was defined. A hexagonal array, corresponding to the size of the SOM array, was then defined along the subspace, its centroid coinciding with that of the mean of the $x(t)$, and the main dimensions of the array being the same as the two largest eigenvalues of the covariance matrix of $x(t)$. The initial values of $m_i(0)$ were taken from these array points. [1]

Our aim in this analysis was to study in what context the "keys" (middle parts in the triplets) occur. The mapping of the $x(t)$ vectors to the SOM was determined by the whole vector $x(t)$, but after learning we labeled the map units according to the middle parts of the $m_i(t)$. In other words, when we compared the "key" parts of the different $m_i(t)$ with a particular word in the list of the selected 150 words (the most frequent ones), the map unit that gave the best match in this comparison was labeled by the said word. It may then also be conceivable that in such a study we should also use only such inputs $x(t)$ for training that have one of the 150 selected words as the "key" part.

In order to equalize the mapping for the selected 150 words statistically and to speed up computation, a solution used in [4] was to average the contexts relating to a particular "key".

In other words, if the input vector is expressed formally as $x = [x_i^T, x_j^T, x_k^T]^T$ where T signifies the transpose of a vector, and x_j is the "key" part, then the true inputs in the "accelerated" learning process were $[E\{x_i^T|x_j\}, 0.2x_j^T, E\{x_k^T|x_j\}]^T$, where E now denotes the (computed) conditional average. (The factor 0.2 in front of x_j^T was used to balance the parts in the input vectors.) In this way we would only have 150 different input vectors that have to be recycled a sufficient number of times in the learning process. The information about all the 7624 words is anyway contained in the conditional averages. Although the above method already works reasonably well, a modification of "averaging" based on auxiliary SOMs was used in this work. For each codebook vector we assigned a small, 2 by 2 SOM that was trained with the input vectors made from the due word triplets. After training, each codebook vector in one small map described more specifically what context was used on the average with that "key" word.

The computation of the map was made in two separate runs. The map was first pretaught using the CNAPS, a massively parallel neurocomputer with fixed-point arithmetic. Preteaching consisted of 600 000 learning cycles. During this run we could use a large radius of N_c. After that, teaching was continued at higher accuracy using a workstation with floating-point arithmetic. During this run 400 000 learning cycles were used.

2.4 Results

The results of the computation are presented in Figure 1. The positions of the words on the map are solely based on the analysis of the contexts performed by the SOM. Explicit lexical categorization of the words was made following the Collins Cobuild Dictionary [7] to help the reader in evaluation of the results. The general organization of the map reflects both syntactical and semantical categories. The most distinct large areas consist of verbs in the top third of the map, and nouns in the bottom right corner.

All verbs can be found in the top section whereas the nouns are located in the lower right corner of the map. In the middle there are words of multiple categories: adverbs, pronouns, prepositions, conjunctions, etc. Modal verbs form a collection of their own among the verbs. Connected to the area of nouns are the pronouns. The three numerals in the material form a cluster. The lexeme "one" is separated from "two" and "three" to some extent, which can be explained by its multiple meaning. Among the verbs the past-tense forms are separated from the present-tense forms and located in the top right corner. A distinct group of possessives can also be found. Even the possessive form "king's" is among them having rather similar contexts. The plain noun "king" is situated within the animate nouns.

Formation of syntactic categories on the map can be explained by sentential context. The context of a word is, quite naturally, dependent on the syntactical restrictions that govern the positions of the words in the text. There may exist many syntactic categories relating to a short context of a word, and therefore, the presence of various categories in the context is only statistical.

Inside the large, syntactically based groups on the map, fine structures of semantic relationships can also be discerned. Consider, for example, the set of nouns. The animate and inanimate nouns form a group of their own each. A set of closely related pairs can be found: "father-mother", "night-day", "child-son", "forest-tree", "head-eyes", and "woman-man".

In addition, some anomalies are present on the map, at the first sight at least. The few adjectives are somewhat scattered on the map. Especially the word "little" has an almost singular location. The reason may lie in the specific uses of that word in phrases like "little by little", or "she walked a little farther".

3 Conclusion

The experimentally found linguistic categories, being determined only implicitly by the self-organizing map, seem to correlate with the categories of concepts occurring in actual cognitive processes. It may also be argued realistically that, in human language learning, naming of explicit syntactic relations is not needed. Expressions heard in the proper context may be sufficient for creating a working model of language.

Contrary to this, the trend in theoretical linguistics has been to construct explicit symbolic models. Comparison of symbolic structures is easier for human linguists; however, the implicitness of "neural" models may turn into an advantage in practical applications where the sharp-bordered symbolic models for semantic processing may too "rough" to meet the requirements of the semantic mapping especially when unrestricted natural language is considered. Linguistic categories can thus be viewed as approximations (generalizations). This is reflected in the emergence of categories in a self-organizing process.

The present study analyzed lexical items of a finite corpus on the basis of a very simple, statistically defined average context. Another approach would be to use complete expressions as input which would give rise to a more detailed analysis of phenomena like polysemy and impreciseness.

Contextual maps can be utilized as a central component in applications like information retrieval and machine translation. The present study concentrates on the analysis of a particular corpus. The method in itself is generally applicable to processing of any textual material or even a combination of text and other modalities.

References

[1] Teuvo Kohonen. *Self-Organizing Maps.* Springer, 1995.

[2] Risto Miikkulainen. *DISCERN: A Distributed Artificial Neural Network Model of Script Processing and Memory.* PhD thesis, Computer Science Department, University of California, Los Angeles, 1990. (Tech. Rep UCLA-AI-90-05).

[3] Risto Miikkulainen. *Subsymbolic Natural Language Processing: An Integrated Model of Scripts, Lexicon, and Memory.* MIT Press, Cambridge, MA, 1993.

[4] Helge Ritter and Teuvo Kohonen. Self-organizing semantic maps. *Biological Cybernectis*, 61(4):241–254, 1989.

[5] J. C. Scholtes. Kohonen feature maps in natural language processing. Technical report, Department of Computational Linguistics, University of Amsterdam, March 1991.

[6] J. C. Scholtes. *Neural Networks in Natural Language Processing and Information Retrieval.* PhD thesis, Universiteit van Amsterdam, Amsterdam, the Netherlands, 1993.

[7] John Sinclair, editor. *Collins Cobuild English Language Dictionary.* Collins, London and Glasgow, 1990.

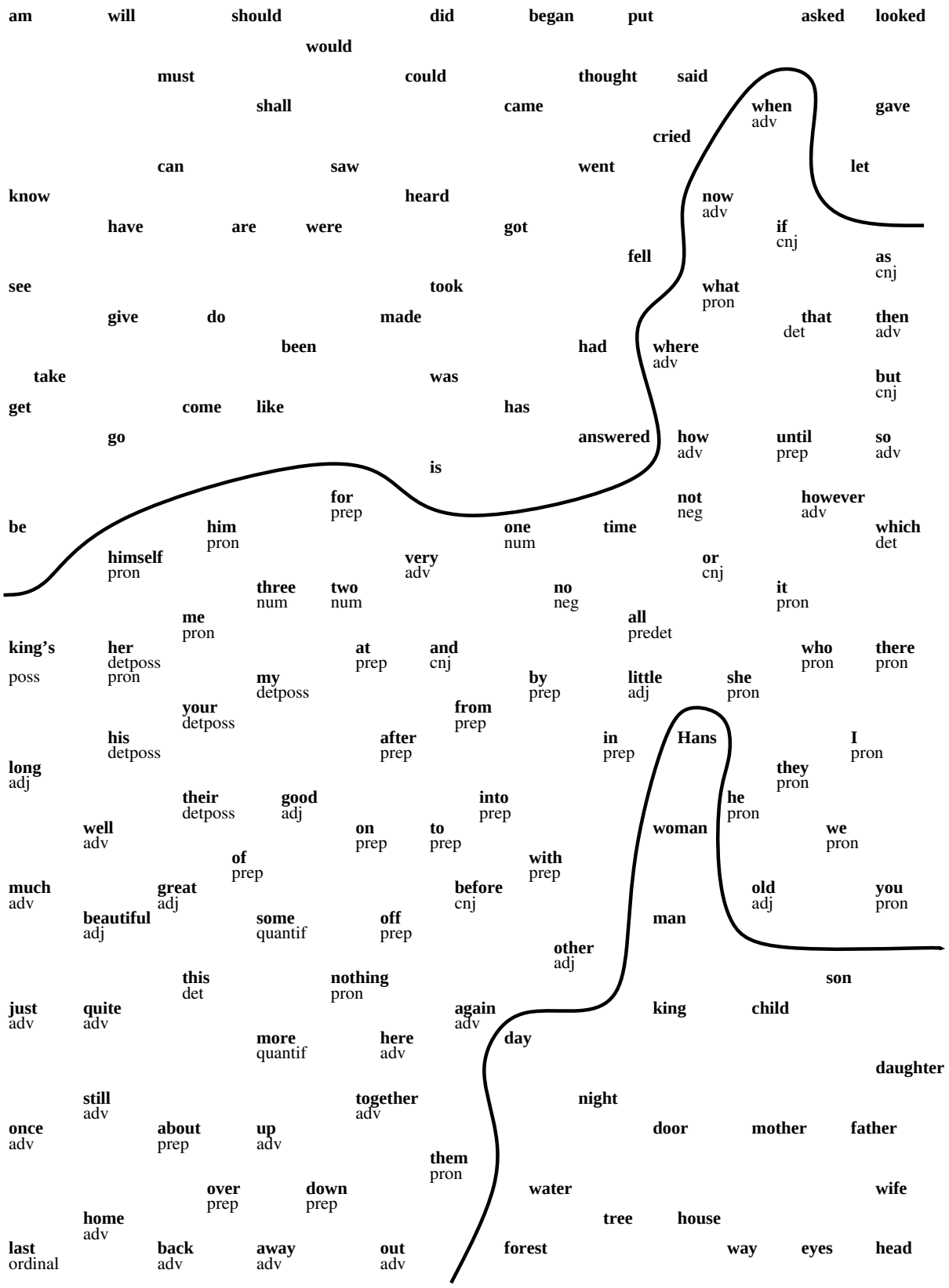

Figure 1: The 150 most frequent words of the Grimm tales, their statistical contextual relations being represented two-dimensionally by the SOM. The words are shown in their due position in the array; no symbols for "neurons" have been drawn. Linguistic categories other than verbs and nouns have been assigned to the words. Many words are ambiguous but usually only the most common category relating to the tales is presented.