EXTRACTING MOST FREQUENT CROATIAN ROOT WORDS USING DIGRAM COMPARISON AND LATENT SEMANTIC ANALYSIS

Zvonimir Radoš
Ph.D. student at the Faculty of Electrical Engeeniring, University of Josip Juraj Strossmayer, Kneza Trpimira 2b, Osijek, Croatia

Dr. Franjo Jović, Josip Job
Faculty of Electrical Engeeniring, University of Josip Juraj Strossmayer, Kneza Trpimira 2b, Osijek, Croatia

Keywords: Morphological analysis, LSA, word tree, stem, root word, knowledge-free

Abstract: A method for extracting root words from Croatian language text is presented. The described method is knowledge-free and can be applied to any language. Morphological and semantic aspects of the language were used. The algorithm creates morph-semantic groups of words and extract common root for every group. For morphological grouping we use digram comparison to group words depending on their morphological similarity. Latent semantic analysis is applied to split morphological groups into semantic subgroups of words. Root words are extracted from every morpho-semantic group. When applied to Croatian language text, among hundred most frequent root words, produced by this algorithm, there were 60 grammatically correct ones and 25 FAP (for all practical purposes) correct root words.

1 INTRODUCTION

Natural language processing (NLP) systems represent one of the key areas of research in artificial intelligence. Use of this kind of systems is wide: from automatic translators and data mining agents to voice commanding robots. Efficiency of NLP system greatly depends on its capability to "understand" the language. Every natural language is composed of its form (verbal language expression and written form) and its semantic content. One of the basic tasks of NLP systems is morphological analysis because computer works only with language expression and uses it to grasp its content. The problem of finding the set of most frequent Croatian roots of words in a given text is elaborated in this work. If one knows the root of the word then he/she can abstract its content since words that share common root usually also share common content.

Main part of the method is morphological analysis where we use the written form of the language to create groups of morphologically similar words. There are cases in language where words with similar written forms have totally opposite content so it is necessary to include semantics in this process of word grouping. With the use of semantics we divide groups of morphologically similar words into subgroups of semantically similar words. As a result we have groups of words that probably share common root word.

The morphological analysis is based on a digram comparison and for the semantic part we use a method called *latent semantic analysis*.

In this paper we took an engineering approach to the problem. The goal was not to extract strictly grammatically correct root words but to define groups of words that share common root.

2 METHOD DESCRIPTION

Algorithm described here consists of several steps shown in figure 1. The input of the algorithm is a plain text in any natural language and the output is a list of most frequent root words in a given text.

Radoš Z., Jović F. and Job J. (2005).
EXTRACTING MOST FREQUENT CROATIAN ROOT WORDS USING DIGRAM COMPARISON AND LATENT SEMANTIC ANALYSIS.
In *Proceedings of the Seventh International Conference on Enterprise Information Systems*, pages 370-373
DOI: 10.5220/0002551903700373

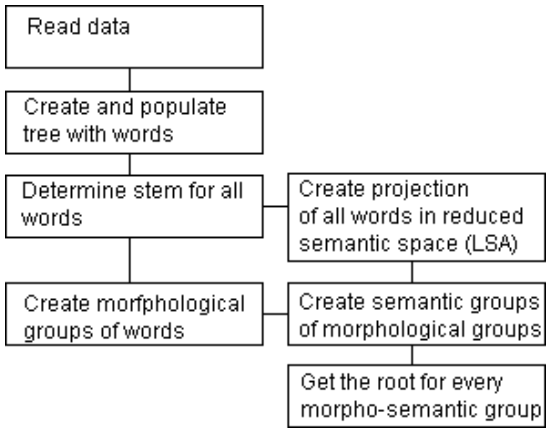

Figure 1: Diagram of the root extraction algorithm

In the first step of the root extraction the algorithm determines the possible stem for every word. The given text is being read word by word and those words are used to populate the tree. Every letter of the word is one node in the tree.

We determine possible affixes from the tree. A possible affix is every branch after the last branching in the tree. These affixes are then ordered by the product of the logarithm of their frequency and their length, i.e.

$$\log(\text{frequency}(\text{affix}_{i-1})) \cdot \text{length}(\text{affix}_{i-1}) < \log(\text{frequency}(\text{affix}_{i})) \cdot \text{length}(\text{affix}_{i}) < \log(\text{frequency}(\text{affix}(x_{i+1})) \cdot \text{length}(\text{affix}_{i+1}) \quad [1]$$

In this way we point out more frequent affixes because the possibility of their correctness is bigger. We also emphasize the longer affixes because their removal gives us "cleaner" stems. This is done twice, once for suffixes and once for prefixes. We just reverse the order of letters in words for prefixes and then populate the tree.

We subtract the first matching affix in a suffix and in a prefix list from every word in the text and rest is a candidate stem for a given word.

2.1 Morphological analysis

The crucial part of this algorithm and also the main part of a morphological analysis is the clustering of words. In this process we group all words from the given text using morphological similarity of their stems. The method used here is a digram comparison (De Roeck, A., W. Al-Fares, 2000). Every stem is divided into pairs of letters with intersect of one letter. Morphological similarity coefficient (SC) is then given by Dice's equation:

$$SC = 2\frac{\text{number of shared unique digrams}}{\text{sum of unique digrams in both stems}}$$

Table 1: Calculation of SC for two words with similar stems

word	stem	Digrams
izgled	*gled*	*gl le ed* (3)
ogledalo	*gleda*	*gl le ed da* (4)
common digrams		*gl le ed* (3)
SC		2·3/(3+4) = **0.86**

The SC coefficient is used for word grouping in clusters of morphologically similar stems. Two words belong to the same stem group if their SC is higher then the set threshold (we've used the threshold 0.65. A higher threshold creates too much morphological groups and a lower threshold decreases the accuracy of the created ones).

2.2 Semantic analysis

So far we have examined only the written form of the language but we need to include semantics in the process for the complete analysis. The proposed method doesn't use semantics by linking expressions with their contents. It rather links together the expressions of words that have similar contents. This is done by using latent semantic analysis (LSA) (T. Laundauer, S. Dumais,).

LSA is a method that is primarily used in classification and retrieval of documents. The original idea is to find some hidden relations between words based on their co-occurrence in a text and use them to retrieve documents relative to the user query.

LSA is used here in a way similar to that described in (P. Schone, D. Jurafsky, 2000.). We create a vector for every word and project it in a

semantic space. If words end up closely we assume that they are related to the similar subject. We create stem-stem matrix $M(i, Np + j)$ where rows are vectors that represent N most frequent stems in a semantic space consisted of $2N$ dimensions. The indices i and j are N most frequent stems (in this algorithm $N = 1000$). The value of p is 0 if a word with the j stem has a positional offset to a word with the i stem in a [-50, -1] range and p is 1 when the offset is in a [1, 50] range. The algorithm takes a word by word from the text and populates the values of the matrix.

The produced matrix gives us co-occurrence frequencies of N most frequent stems in the text. In order to emphasize these frequencies the z transformation is used:

$$z_{ij} = \frac{M_{ij} - \mu_i}{\sigma_i}, \quad [3]$$

where μ_i is a mean value and σ_i a standard deviation of the i^{th} row, M_{ij} is an original frequency and z_{ij} is a new transformed frequency of co-occurrence of the i^{th} stem and the j^{th} stem.

We perform singular value decomposition (SVD) on the matrix with transformed frequencies. This decomposition transforms the original semantic space into a new space with dimensions ordered by their relevance. Now we can reduce the number of dimensions with a minimum loss of information. This reduction of dimensions reveals the hidden semantic correlations between words and causes the words with similar meaning to be projected closely in this reduced semantic space. When SVD is applied to the M matrix we get:

$$M = USV^T, \quad [4]$$

where U and V^T are orthogonal matrices and S is the diagonal matrix composed of singular values of M matrix. If we the create U_{Nxk} and V_{Nxk} matrices using first k columns of U and V matrices and S_{kxk} matrix using first k rows and columns of S matrix, we can create projection of the M matrix in a reduced space consisted of k dimensions using equation:

$$M_{Nxk} = U_{Nxk} S_{kxk} \quad [5]$$

Singular values in the S matrix are ordered in a descending order and they are considered to be weights for relevance of a particular dimension: $S(1,1)$ for the dimension 1, $S(2,2)$ for the dimension 2 and so on. Now we have the vectors in this new and reduced space for every of the N most frequent stems. The rest of the stems are folded in (T. Laundauer, S. Dumais, 1997). First, we create a vector that indicates frequency of co-occurrence of that stem and the N most frequent stems (the same as the initial creation of the M matrix). To get the projection of this vector in the reduced space we multiply it with the V_{2Nxk} matrix.

The semantic similarity of two words is then given by position of their stems in the semantic space. We use the cosine of the angle between vectors as a measure of similarity. Using this similarity we can now divide every group of morphologically similar words (created using digram comparison) into subgroups of semantically close words. This way, we have obtained groups of words from which we can extract each root word without any predefined knowledge about a particular language.

2.3 Determining the common root

The last part of this method determines the root word for every group. Here we use the simplest possible method and the algorithm takes the string that is present in stems of all the words in the group for root of the words in that group. The assigned root is then compared to the list of already obtained roots and if shorter root exists, we take that shorter one as valid and discard the longer root. After all groups of stems have been processed we order the root list according to the number of words in which they can be found.

3 RESULTS

The described method has been developed and applied to a Croatian language text. The Croatian language possesses a lot of flexes that have a huge impact on written form of the language. That is why Croatian language presents a big challenge for the natural language processing systems.

The text used in the analysis has been composed from the three thematically different documents and has 41000 words in total, among which were 6100 different ones. The first part of the algorithm created 1834 morphological groups of words. There were many single or double-word groups. The reason for this is a quite small text with insufficient number of different lexical forms of one word.

When we extracted the roots directly from the morphological groups we got 2094 root words among which there were 723 derived from single-word groups. When we used semantics as a correcting tool for morphological analysis we got 3041 root words and among them 1651 that were from single-word groups.

The accuracy of the algorithm is being tested through the percentage of the correct roots among a

hundred of the most frequent ones. All produced roots can be divided into three categories: the grammatically correct ones, correct for all practical purposes (*FAP*) and incorrect. *FAP* correct roots are those that are not grammatically correct but have been extracted from the group of words that share common root. They are too long to be grammatically correct, but with some improvements in the last step of the root extraction algorithm, they could become grammatically correct. On the other hand, incorrect roots are those that are extracted from the groups of words that don't share a common root.

The results are shown in Table 2.

Table 2: Accuracy of hundred most frequent root words extracted using morphological and semantic analysis

Grammatically correct	60%
FAP correct	25%
Incorrect	15%

With the increase of a word corpus, the growth of the number of morpho-semantic groups would slow down and the average number of words per one group would be bigger. This would increase the number of grammatically correct ones and decrease the number of FAP ("for all practical purposes") correct root words. For eliminating the incorrect root words we need to make some improvements in clustering of morphologically similar words. One way would be to produce a positional weight function that would stress the digrams at the beginning, end or middle of the stem, depending on the processed language. It should also be allowed that one word be assigned to more groups, but then the last step of the algorithm (extracting root words from morpho-semantic groups of words) should be improved because proposed one it is too rigid. To achieve that, some sort of weighting could also be used.

The same text was processed using Goldsmith's *Linguistica* (a tool for morphological analysis). Since *Linguistica* is more oriented on the stem of the word, root words produced by it were too long (prefixes were rarely removed). Among hundred most frequent roots there were: 15 grammatically correct ones, 75 *FAP* correct ones and 10 incorrect ones. Since both methods greatly depend on the size of the corpus more extensive tests are needed.

4 CONCLUSION

The goal of this paper is to explore methods for a complete (morphological and semantic) knowledge free computer analysis of any natural language text. The method described here gave good results even when applied to a small text where only a few lexical variants of every word are present. Further improvements of the algorithm are necessary to avoid creation of the incorrect morpho-semantic groups of words. Knowledge-free tools for morphological analysis is probably the right choice for languages that are not world-wide spread (as English is), because creation of morphological dictionaries for "local" languages has questionable cost effectiveness.

REFERENCES

F. C. Graham, 2004. Large Dynamic Graphs: What Can Researchers Learn From Them?, *SIAM News, vol. 37., no. 3.*

T. Laundauer, S. Dumais, 1997. A Solution to Plato's Problem, The Latent Semantic Analysis Theory of Acquisition, Induction and Representation of Knowledge, *Psychological Review*, no. 104., pp. 211-240.

P. Schone, D. Jurafsky, 2000. Knowledge-Free Induction of Morphology Using Latent Semantic Analysis, *Proceedings of CoNLL-2000 and LLL-2000*, Lisbon, Portugal, pp. 67-72.

De Roeck, A., W. Al-Fares, 2000. A Morphologically Sensitive Clustering Algorithm for Identifying Arabic Roots*, Proceedings of the 38th Annual Meeting of the ACL*, Hong Kong.

R. Scitovski, 1999. *Numerička Matematika*, Elektrotehnički fakultet Osijek, Osijek.

P. Nakov, A. Popov, P. Mateev, 2001. Weight Functions Impact on LSA Performance, *EuroConference RANLP'2001, Tzigov Chark*, Bulgaria, pp. 187-193.

C. D. Manning, H. Schűtze, 1999. Foundations of Statistical Natural Language Processing, *MIT Press*, Cambridge, MA, pp. 554-566.

M. Moguš, M. Bratanić, M. Tadić, 1999. *Hrvatski čestotni rječnik*, Školska knjiga, Zagreb.

J. Goldsmith, 2001. Unsupervised Learning of the Morphology of a Natural Language. *Computational Linguistics*. 153-189.