STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **XLVIII**, Number 2, 2003

HIERARCHICAL CLUSTERING ALGORITHMS FOR REPETITIVE SIMILARITY VALUES

DANA AVRAM LUPŞA, GABRIELA ŞERBAN, AND DOINA TĂTAR

Abstract. This paper presents a novel variant of the hierarchical clustering from [2]. We tried to solve the problem of repetitive similarity values that appears on distributional similarity values. Also we propose an algorithm to build a similarity tree as a taxonomy that respects the hierarchical clusters determined above.

1. Introduction

Bootstrapping semantics from text is one of the greatest challenges in natural language learning. Clustering nouns can be useful in construction of a set of synonyms for word sense disambiguation, to perform query expansion in QA systems [9], to build ontology from a text, in data mining, etc., especially for languages others than English, for which doesn't exist a hierarchy such as WordNet (as in Romanian language case). One very surprising approach is an unsupervised algorithm that automatically discovers word senses from text.

Automatic word sense discovery has applications of many kinds. It can greatly facilitate a lexicographer's work and can be used to automatically construct corpus-based similarity trees or to tune existing ones.

We study distributional similarity measures for the purpose of improving some noun clustering methods [2]. We suggest two algorithms that obtain clusters and similarity trees for nouns. Starting with hierarchical clustering algorithm, we consider the case when the similarity values can repeat and suggest a method to determine the taxonomy with respect of hierarchical clusters found by the hierarchical clustering algorithm.

This paper is organized as follows. In section 2, we present some methods that extract words similarity from untagged corpus. A comparison among the precision of the results is also made. Section 3 describes the agglomerative algorithm for hierarchical clustering and it's modified version. Some experimental results are also shown. In section 4, we present the novel agglomerative algorithm for similarity tree. We outline the similarity between the clustering algorithm and the similarity

Received by the editors: October 15, 2003.

2000 *Mathematics Subject Classification.* 62H3, 68Q25, 65Q55, 68R10, 68T50.

1998 *CR Categories and Descriptors.* I.2.7 [**Artificial Intelligence**]: Natural Language Processing – *Text analysis* ; I.5.4. [**Pattern Recognition**]: Applications – *Text processing* .

tree for the experimental results considered. Finally, section 5 sketches applications of the algorithm and discusses future work.

2. Word similarities

Semantic knowledge is increasingly important in NLP. The key of organizing semantic knowledge is to define reasonable similarity measures between words. In many papers the similarity between two words is obtained by the n-grams models [11], by mutual information [3] or by syntactic relations [13]. One other way to define this similarity is the vector space model [5, 12, 7] which we use in this paper. The idea of vector-based semantic analysis is to understand the meaning of a word one has to considering its use in the context of concrete language behavior. The distributional pattern of a word is defined by the contexts in which the word occurs, where context is defined simply as an arbitrarily large sample of linguistic data that contains the word in question.

Syntactic analysis provides some potentially relevant information for clustering [10]. For a corpus in Romanian language the relation predicate-object or subject-predicate can be estimated after position: the object is almost always after the predicate, the subject is before the predicate. So we replaced a syntactical analysis by constructing context vectors as in **Definition 2**.

The reason for using narrow context windows as opposed to arbitrarily contexts is the assumptions that the semantically most significant context is the immediate vicinity of a word. That is, one would expect the words closest to the focus word to be of greater importance than the other words in the text.

Definition 1. *In* **AlgUnord** *algorithm* ([2]) *the vector*
$\vec{w_i} = (w_i^1, w_i^2, \cdots, w_i^m)$
is associated with a noun w_i *as following: let us consider that* $\{v_1, v_2, \cdots, v_m\}$ *are* m *verbs of a highest frequency in corpus. We define:*

$$w_i^j = number\, of\, occurences\, of\, the\, verb\, v_j\, in\, the\, same\, context\, with\, w_i$$

Let us remark that other vector-space models were used in the literature. For example, in [1] is presented a hierarchy of nouns such that the vector $\vec{w_i} = (w_i^1, w_i^2, \cdots, w_i^m)$ associated with a noun w_i is constructed as follows: $w_i^j = 1$, if the noun w_j occurs after w_i separated by the conjunction *and* or an appositive, or else $w_i^j = 0$.

Definition 2. *In* **AlgOrd** *algorithm* ([2, 5]) *the vector* $\vec{w_i}$ *is associated with a noun* w_i *as following: for each verb* v_j *is calculated a sub-vector* $(v_j^{-3}, v_j^{-2}, v_j^{-1}, v_j^{+1}, v_j^{+2}, v_j^{+3})$ *where* v_j^{-3} *=1 if* v_j *occurs in a windows context of* w_i *in the position -3 or* v_j^{-3} *=0 else, and so far for* $v_j^{-2}, v_j^{-1}, v_j^{+1}, v_j^{+2}, v_j^{+3}$.

Finally, the vector $\vec{w_i}$ *is obtained by the concatenation, in order, of all sub-vectors of verbs* $\{v_1, v_2, \cdots, v_m\}$.

Let us remark that in **AlgOrd** the number of components of the noun's vector $\vec{w_i}$ is $6 \times m$, while in **AlgUnord** is m. The dimension of a window can be 4 (so the subvectors for a verb v_j are $v_j^{-2}, v_j^{-1}, v_j^{+1}, v_j^{+2}$) or 2 (and the subvectors are: v_j^{-1}, v_j^{+1}). We will denote the windows in each case by 3+3, 2+2 or 1+1.

In both algorithms, if a noun w_i occurs in more contexts, the final vector $\vec{w_i}$ is obtained as the average of all the context vectors.

Let us observe that the corpus does not have to be POS tagged or parsed and that one can use a stemmer to recognize the flexional occurrences of the same word (Romanian language is a very inflexional language).

Let us consider that the objects to be clustered are the vectors of n nouns, $\{w_1, w_2, \cdots, w_n\}$ and that a vector is associated with a noun w_i as above.

The similarity measure between two nouns w_a, w_b is the *cosine* between the vectors $\vec{w_a}$ and $\vec{w_b}$ [6]:

$$cos(\vec{w_a}, \vec{w_b}) = \frac{\sum_{j=1}^{m} w_a^j \times w_b^j}{\sqrt{\sum_{j=1}^{m} {w_a^j}^2} \times \sqrt{\sum_{j=1}^{m} {w_b^j}^2}}$$

and the distance (dissimilarity) is $d(\vec{w_a}, \vec{w_b}) = \frac{1}{cos(\vec{w_a}, \vec{w_b})}$.

In **Table 1** we present, comparatively, the precision of the clustering algorithms for our clustering experiment.

	AlgOrd (3+3)	AlgUnord
non-hierarchical	63%	54%
hierarchical	45%	36%

TABLE 1. Precision of clustering algorithms for the proposed experiment

In the followings, we will consider the results of the studied hierarchical algorithms (see Table 1). The decision was made to support the study of repetitive similarity values. The similarity values are repetitive more significant for the hierarchical algorithm than for the non-hierarchical ones.

The distributional similarity matrices obtained for the Romanian words: *asociatie, durata, localitate, oameni, oras, organizatie, partid, persoana, perioada, sat, timp* by the considered hierarchical algorithms are presented in **Table 2** and **Table3**. For readability reasons the values shown are rounded to 9 decimal characters.

The similarity values are repetitive, as shown in the **Fig 1**.

In what follows we will give an algorithm for hierarchical clustering, that handle repetitive values.

3. New hierarchical clustering algorithm

Word clustering is a technique for partitioning sets of words into subsets of semantically similar words and is increasingly becoming a major technique used in

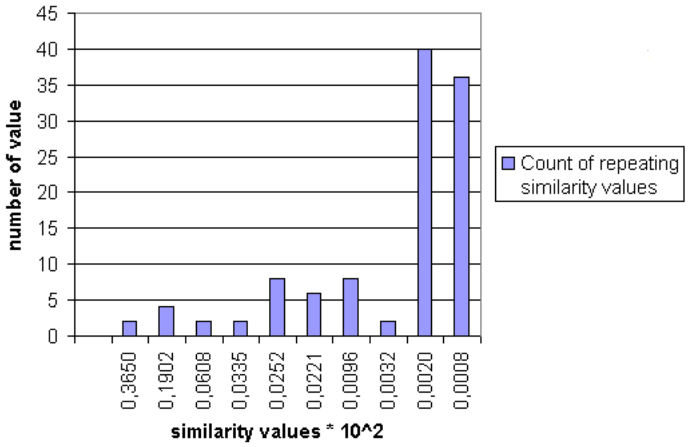

FIGURE 1. Repetitive similarity values obtained by hierarhical algorithm **AlgUnord**

a number of NLP tasks ranging from word sense or structural disambiguation to information retrieval and filtering. In the literature [4], two main different types of similarity have been used. They can be characterized as follows:

1. *paradigmatic or substitutional similarity*: two words that are paradigmatically similar may be substituted one for another in a particular context. For example, in the context *I read the book* , the word *book* can be replaced by *magazine* with no violation of the semantic well-formedness of the sentence, and therefore the two words can be said to be paradigmatically similar;

2. *syntagmatic similarity*: two words that are syntagmatically similar significantly occur together in text. For instance, *cut* and *knife* are syntagmatically similar since they typically co-occur within the same context.

Both types of similarity, computed through different methods, are used in the framework of a wide range of NLP applications.

The agglomerative algorithm for hierarchical clustering that we intend to use is part of the second category. The original hierarchical clustering algorithm [2, 6] is described in what follows.

Agglomerative algorithm for hierarchical clustering

Input

The set $X = \{w_1, w_2, \ldots, w_n\}$ of n words to be clusterised,
the similarity function $sim : X \times X \to R$.

Output

The set of hierarchical clusters

$C = \{C_1^0, C_2^0, \ldots, C_j^n\}$

BEGIN
FOR $i := 1$ TO n DO
$C_i^0 := w_i$
ENDFOR
$step := 0$
$C^0 := \{C_1^0, C_2^0, \ldots, C_n^0\}$
$C := C^0$
WHILE $|C| > 1$ DO
$step := step + 1$
$C^{<step>} := C^{<step-1>}$
$(C_{u*}^{<step>}, C_{v*}^{<step>}) :=$
$argmax_{(C_u^{<step>}, C_v^{<step>})} sim(C_u^{<step>}, C_v^{<step>}),\ u <> v$
$C_*^{<step>} := C_{u*}^{<step>} \cup C_{v*}^{<step>}$
$C^{<step>} := (C^{<step>} \setminus \{C_{u*}^{<step>}, C_{v*}^{<step>}\}) \cup C_*^{<step>}$
$C := C \cup C^{<step>}$
ENDWHILE
END

As similarity $sim(C_u, C_v)$ we considered *average -link* similarity:

$$sim(C_u, C_v) = \frac{\sum_{a_i \in C_u} \sum_{b_j \in C_v} sim(a_i, b_j)}{\mid C_u \mid \times \mid C_v \mid}.$$

Taken as input the similarities from **Table 2**, the resulting hierarchical clusters are shown in **Fig 2** . The circles indicate the clusters at a certain moment and the numbers indicate the step when the cluster was formed.

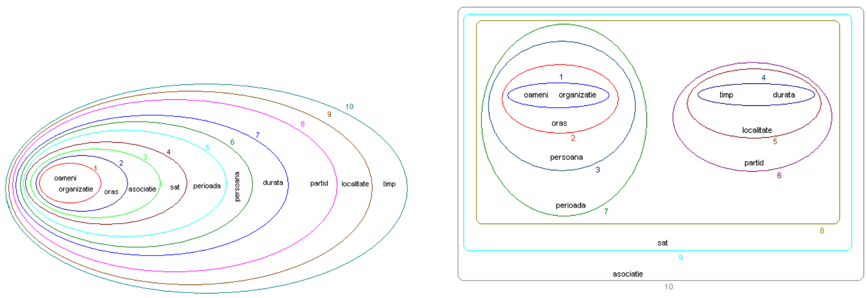

FIGURE 2. Results of agglomerative algorithm for hierarchical clustering on experimental data set (table 2 and 3)

When the similarity values have many repetitive values, as shown in **Fig 1**, it could be possible that the similarity between different clusters is the same. The idea behind the new hierarchical clustering algorithm is to consider at each step all the clusters that are closest to each other, as the similarity value is showing. The new algorithm and some experimental results are presented in what follows.

Agglomerative algorithm for hierarchical clustering and repetitive similarity values

Input
The set $X = \{w_1, w_2, \ldots, w_n\}$ of n words to be clusterised,
the similarity function $sim : X \times X \rightarrow R$.
Output
The set of hierarchical clusters
$C = \{C_1^0, C_2^0, \ldots, C_{n_k}^k\}$

BEGIN
FOR $i := 1$ TO n DO
$C_i^0 := w_i$
ENDFOR
$step := 0$
$C^0 := \{C_1^0, C_2^0, \ldots, C_n^0\}$
$C := \{C^0\}$
WHILE $|C| > 1$ DO
$step := step + 1$
$C^{<step>} := C^{<step>-1}$
$smax := max_{(C_u^{<step>}, C_v^{<step>})} sim(C_u^{<step>}, C_v^{<step>})$
FOR each $(C_u^{<step>}, C_v^{<step>}) \in C \times C$, $u <> v$
IF $smax := sim(C_u^{<step>}, C_v^{<step>})$
$C_*^{<step>} := C_u^{<step>} \cup C_v^{<step>}$
$C^{<step>} := C^{<step>} \setminus \{C_u^{<step>}, C_v^{<step>}\} \cup C_*^{<step>}$
END IF
END FOR
$C := C \cup C^{<step>}$
ENDWHILE
END

Taken as input the similarity from table **Table 2** and **Table 3**, with higher rate repetitive value, the results are shown in **Fig 3**.

4. Algorithm to create a similarity tree with respect to hierarchical clusters

Lexical semantics relations play an essential role in lexical semantics and interfere in many levels in natural language comprehension and production. They are also a central element in the organization of lexical semantics knowledge bases.

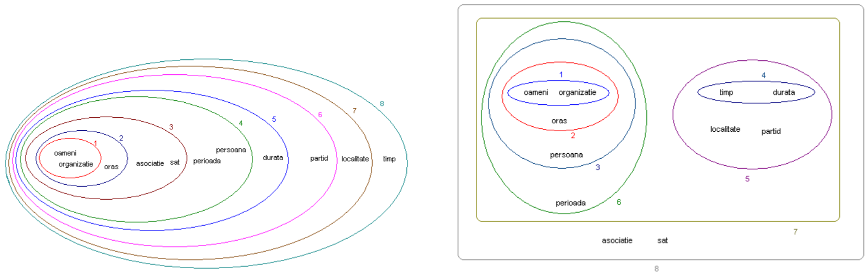

FIGURE 3. Results of agglomerative algorithm for hierarchical clustering on repetitive similarities on experimental data set (table 2 and 3)

Two words W1 and W2 denoting respectively sets of entities E1 and E2, are in one of the following four relations [4]:

identity: E1 := E2,
inclusion: E2 is included into E1,
overlapp: E1 and E2 have a non-empty intersection,
but one is not included into the other,
disjunction: E1 and E2 have no element in common.

These relations support various types of lexical configurations such as the type/subtype relation.

We are interested in constructing a tree structure among similar words so that different senses of a given word can be identified with different subtrees [8]. In what follows we try to model the hierahical clustering algorithm to extract such tree hierarchical structure that we call similarity trees or taxonomy.

For the similarity tree, unification of two clusters in the hierarchical algorithm means to establish a link between two words from the two clusters that are the most similar . The question is now: how to choose those two words when similarity values between words are highly repetitive.

The solution is to find a way to filter the words from a cluster in order to get only one.

The filters we propose are:

- Filter 1: word of maximum similarity
  - choose among candidate words in the two clusters the pairs that have maximum similarity among all pairs of words
- Filter 2: most important words in the cluster
  - choose among candidate words in the two clusters the words that have the sum of the similarities with the other words in the cluster maximum
- Filter 3: most important words for the new cluster

  - choose among candidate words in the two clusters the words that have the sum of the similarities with all the other words in the two clusters maximum
- Filter 4: most important words for the entire set
  - choose among candidate words the words that have the sum of the similarities with all the other words in the entire set maximum

If all those cannot identify a singular word, this indicates that similarity value sets have too many repetitive values that cannot make a distinction among words in some groups. Filtering can be repeatedly applied by using other similarity values sets if it does not obtain an unique word.

Filter algorithm

Input
  $CW1 = \{cw_{11}, cw_{12}, \ldots\}$ the set of words to be filtered
  $CW2 = \{cw_{21}, cw_{22}, \ldots\}$ a set of words distinct to CW1
  W : a set of words so that CW1 and CW2 are part of it
    (the set of all considered words)
  $sim : W \times W \rightarrow R$ the similarity function
Output
  $CW1 = \{cw', cw'', \ldots\}$: the filtered CW1

BEGIN
  IF $|CW1| > 1$ /*** filter 1 ***/
    $msim1 := max\{sim(c1, c2) \mid c1 \in CW1, c2 \in CW2\}$
    $CW1 := \{c1 \mid \exists c2 \in CW2 \ so\ that\ msim1 = sim(c1, c2)\}$
  ENDIF
  IF $|CW1| > 1$ /*** filter 2 ***/
    $msim2 := max\{\sum_{cw2} sim(cw1, cw2) \mid$
      $cw1 \in CW1, cw2 \in CW1, cw1 <> cw2\}$
    $CW1 := \{cw1 \mid msim2 = \sum_{cw2} sim(cw1, cw2),$
      $cw1 \in CW1,\ cw2 \in CW1, cw1 <> cw2\}$
  ENDIF
  IF $|CW1| > 1$ /*** filter 3***/
    $msim3 := max\{\sum_{cw2} sim(cw1, cw2) \mid$
      $cw1 \in CW1, cw2 \in (CW1 \cup CW2, cw1 <> cw2\}$
    $CW1 := \{cw1 \mid msim3 = \sum_{cw2} sim(cw1, cw2),$
      $cw1 \in CW1,\ cw2 \in (CW1 \cup CW2), cw1 <> cw2\}$
  ENDIF
  IF $|CW1| > 1$ /*** filter 4 ***/
    $msim4 := max\{\sum_{cw2} sim(cw1, cw2) \mid$
      $cw1 \in CW1, cw2 \in W, cw1 <> cw2\}$
    $CW1 := \{cw1 \mid msim4 = \sum_{cw2} sim(cw1, cw2),$
      $cw1 \in CW1,\ cw2 \in W, cw1 <> cw2\}$
  ENDIF

END

Agglomerative algorithm for similarity tree

Input
The set $W = \{w_1, w_2, \ldots, w_n\}$ of n words to be clustered,
$S1 : W \times W \rightarrow R$ main similarity function
$S2, \ldots, Sk : W \times W \rightarrow R$ other similarity functions
Output
T similarity tree that respects clusters created by using
agglomerative hierarchical clustering algorithm

```
BEGIN
    T := {}
    FOR i := 1 TO n DO
        C_i := {w_i}
    ENDFOR
    C := {C_1, C_2, , C_n}
    WHILE |C| > 1 DO
        smax := max_(Cu,Cv) sim(Cu, Cv), u <> v
        FOR each (Cu, Cv) ∈ C × C, sim(Cu, Cv)) = smax and u <> v
            FILTER(Cu, Cv , W, S1)
            FILTER(Cv, Cu , W, S1)
            i := 1
            WHILE (i < k) AND (|Cu| > 1 OR |Cv| > 1)
                C'u := Cu
                IF |Cu| > 1
                    FILTER(Cu, Cv , W, Si)
                ENDIF
                IF |Cv| > 1
                    FILTER(Cv, C'u , W, Si)
                ENDIF
                i := i + 1
            ENDWHILE
            IF |Cu| > 1 OR |Cv| > 1
                MESSAGE: "Undecidable"
                END ALGORITHM
            ENDIF
            /* Consider that Cu = {cw1'} and Cv = {cw2'} */
            T := T ∪ (cw1', cw2')
            C := (C \ {Cu, Cv}) ∪ {Cu ∪ Cv}
        ENDFOR
     ENDWHILE
END
```

The algorithm has the advantage of combining the clustering methods with the filetring algorithm in order to obtain similarity trees.

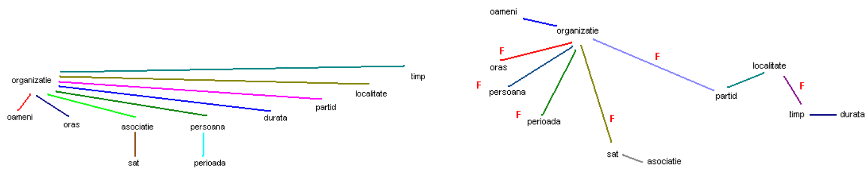

FIGURE 4. Result of agglomerative algorithm for similarity tree on experimental data set in Table 2 and 3 (hierarchical **AlgOrd**)

Let us construct similarity tree starting with the same similarity values set as used for hierarchical clusters. For those similarity values, the taxonomy algorithm needs supplementary similarity values. Taken as supplementary similarities those from nonhierarchical AlgOrd algorithm, the algorithm is decidable and the two similarity trees that are built for the hierarchical clusters presented above, looks like in **Fig 4**. The big "F" symbol in the figures indicates links that were not decidable without filtering.

5. Conclusions and future research

This paper gives two algorithms to determine hierarchical clusters and similarity trees, starting from untagged corpus data.

We intend to use the method of extracting similarity trees from untagged corpus for semiautomatic building of a IS-A hierarchy for Romanian languaage.

Appendix

	asociatie	durata	localitate	oameni	oras	organizatie	partid	perioada	persoana	sat	timp
asociatie	1	0.96707415	0.95188788	0.98411205	0.98411205	0.98411205	0.95686704	0.97812600	0.97812600	0.99181731	0.94460959
durata	0.96707415	1	0.95188788	0.96707415	0.96707415	0.96707415	0.95686704	0.96707415	0.96707415	0.96707415	0.94460959
localitate	0.95188788	0.95188788	1	0.95188788	0.95188788	0.95188788	0.95188788	0.95188788	0.95188788	0.95188788	0.94460959
oameni	0.98411205	0.96707415	0.95188788	1	0.99846577	0.99893616	0.95686704	0.97812600	0.97812600	0.98411205	0.94460959
oras	0.98411205	0.96707415	0.95188788	0.99846577	1	0.99846577	0.95686704	0.97812600	0.97812600	0.98411205	0.94460959
organizatie	0.98411205	0.96707415	0.95188788	0.99893616	0.99846577	1	0.95686704	0.97812600	0.97812600	0.98411205	0.94460959
partid	0.95686704	0.95686704	0.95188788	0.95686704	0.95686704	0.95686704	1	0.95686704	0.95686704	0.95686704	0.94460959
perioada	0.97812600	0.96707415	0.95188788	0.97812600	0.97812600	0.97812600	0.95686704	1	0.99615956	0.97812600	0.94460959
persoana	0.97812600	0.96707415	0.95188788	0.97812600	0.97812600	0.97812600	0.95686704	0.99615956	1	0.97812600	0.94460959
sat	0.99181731	0.96707415	0.95188788	0.98411205	0.98411205	0.98411205	0.95686704	0.97812600	0.97812600	1	0.94460959
timp	0.94460959	0.94460959	0.94460959	0.94460959	0.94460959	0.94460959	0.94460959	0.94460959	0.94460959	0.94460959	1

TABLE 2. Similarity data set obtained for hierarchical **AlgOrd** algorithm

	asociatie	durata	localitate	oameni	oras	organizatie	partid	perioada	persoana	sat	timp
asociatie	1	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	0.00003211	0.00000849
durata	0.00000849	1	0.00025204	0.00002015	0.00002015	0.00002015	0.00025204	0.00002015	0.00002015	0.00000849	0.00060790
localitate	0.00000849	0.00025204	1	0.00002015	0.00002015	0.00002015	0.00033500	0.00002015	0.00002015	0.00000849	0.00025204
oameni	0.00000849	0.00002015	0.00002015	1	0.00190216	0.00364963	0.00002015	0.00009627	0.00022050	0.00000849	0.00002015
oras	0.00000849	0.00002015	0.00002015	0.00190216	1	0.00190216	0.00002015	0.00009627	0.00022050	0.00000849	0.00002015
organizatie	0.00000849	0.00002015	0.00002015	0.00364963	0.00190216	1	0.00002015	0.00009627	0.00022050	0.00000849	0.00002015
partid	0.00000849	0.00025204	0.00033500	0.00002015	0.00002015	0.00002015	1	0.00002015	0.00002015	0.00000849	0.00025204
perioada	0.00000849	0.00002015	0.00002015	0.00009627	0.00009627	0.00009627	0.00002015	1	0.00009627	0.00000849	0.00002015
persoana	0.00000849	0.00002015	0.00002015	0.00022050	0.00022050	0.00022050	0.00002015	0.00009627	1	0.00000849	0.00002015
sat	0.00003211	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	0.00000849	1	0.00000849
timp	0.00000849	0.00060790	0.00025204	0.00002015	0.00002015	0.00002015	0.00025204	0.00002015	0.00002015	0.00000849	1

TABLE 3. Similarity data set obtained for hierarchical **AlgUnord** algorithm

References

[1] S. A. Caraballo, *Automatic construction of hypernym-labeled noun hierarchy from text*, Proceedings of ACL, 1999.
[2] D. Avram Lupşa, G. Şerban, D. Tătar, *From noun's clustering to taxonomies on a untagged corpus*, MPS-Mathematical Preprint Server: Applied Mathematics, 0309004, 2003.
[3] I. Dagan, L. Lee, F. C. N. Pereira, *Similarity-based models of Word Cooccurences Probabilities*, MLJ 34(1-3), 1999.
[4] EAGLES Lexicon Interest Group, A. Sanfilippo, comp., *EAGLES LE3-4244, Preliminary Recommendations on Lexical Semantic Encoding, Final Report*, 1999.
[5] S. Gauch, J. Wang, S. M. Rachakonda, *A corpus analysis approach for automatic query expansion and its extension to multiple databases*, CIKM'97, Conference on Information and Knowledge management, 1997.
[6] C. Manning, H. Schutze, *Foundation of statistical natural language processing*, MIT, 1999.
[7] J. Karlgren, M. Sahlgren, *From words to understanding*, CSLI 2001, pp 294-308, 2001.
[8] D. Lin, *Automatic retrieval and clustering of similar words*, COLING-ACL'98, Montreal, 1998.
[9] C. Oraşan, D. Tătar, G. Şerban, D. Avram, A. Oneţ, *How to build a QA system in your back-garden: application to Romanian*, EACL 2003, Budapest, Hungary, 2003.
[10] V. Pekar, S. Staab, *Word classification based on combined measures of distributional and semantic similarity*, EACL 2003, Budapest, Hungary, 2003.
[11] P. Resnik, *Semantic Similarity in a Taxonomy: An information-Based Measure and its Application to Problems of Ambiguity in Natural language*, Journal of AI Research, 1998.
[12] M. Sahlgren, *Vector-Based Semantic Analysis: Representing Word Meanings Based on Random Labels*, Proceedings of the ESSLLI 2001 Workshop on Semantic Knowledge Acquisition and Categorisation, Helsinki, Finland, 2001.
[13] D. Widdows, *A mathematical model for context and word meaning*, Fourth International Conference on Modeling and using context, Stanford, California, 2003.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Department of Computer Science, Cluj-Napoca, Romania

e-mail adresses: davram@cs.ubbcluj.ro, gabis@cs.ubbcluj.ro, dtatar@cs.ubbcluj.ro