Comparing Measures of Semantic Similarity

Nikola Ljubešić, Damir Boras, Nikola Bakarić, Jasmina Njavro
Department of Information Sciences
Faculty of Humanities and Social Sciences
Ivana Lučića 3, 10000 Zagreb, Croatia
E-mail: {nljubesi,dboras,nbakaric,jnjavro}@ffzg.hr

Abstract *The aim of this paper is to compare different methods for automatic extraction of semantic similarity measures from corpora. The semantic similarity measure is proven to be very useful for many tasks in natural language processing like information retrieval, information extraction, machine translation etc. Additionally, one of the main problems in natural language processing is data sparseness since no language sample is large enough to seize all possible language combinations. In our research we experiment with four different measures of association with context and eight different measures of vector similarity. The results show that the Jensen-Shannon divergence and L1 and L2 norm outperform other measures of vector similarity regardless of the measure of association with context used. Maximum likelihood estimate and t-test show better results than other measures of association with context.*

Keywords. semantic similarity, lexical resources, measures of vector similarity, measures of association with context

1 Introduction

Language resources are basic information sources in language technology. Building such resources manually requires a large amount of linguistic expertise and time. Therefore the possibility of building resources automatically from language samples, ie. corpora is a very appealing one.

The notion of semantic similarity is very broad and differently understood among specialists. Some understand semantic similarity as being equal to synonymy while some assume different semantic relations such as hyponymy and meronymy.

Lexical resources containing information about semantic similarity can be used in many natural language processing tasks. The most popular one is query extraction in information retrieval. Beyond information retrieval, semantic similarity information is a very powerful tool for fighting the general NLP problem of data sparseness.

The automation of extracting semantically similar lexemes is based on the assumption that semantically similar lexemes occur in similar contexts. Therefore the dominant methodology for extracting semantically similar lexemes is taken from information retrieval - every lexeme is represented as a vector containing frequency information of co-occuring lexemes. The semantic distance is computed as the distance between vectors.

There are three basic stages in computing semantic similarity:

1. building co-occurrence vectors
2. measuring association with context
3. measuring vector similarity

These three stages are discussed in the next three subsections.

1.1 Methods for building co-occurrence vectors

Every lexeme of interest is represented as a vector where its dimensions are features like tokens, lemmata or syntactic relations the lexeme co-occurs with [7]. If co-occurence vectors are built using tokens or lemmata, it is common to define two sets - $V1$ and $V2$. The set $V1$ includes all lexemes of interest while $V2$ includes all tokens or lemmata taken

as features. While building co-occurence vectors, co-occurence of features from $V2$ with lexemes from $V1$ is measured.

There are several methods of defining the breadth of the observed context. The most common is the window method. If using the window method, three factors have to be considered [2]:

1. width - how many characters or words the window extends over
2. symmetry - if the window is symmetric
3. boundaries - whether the window is fixed regarding boundaries such as sentence and paragraph breaks

The window method used in this research considers sentence brakes as window boundaries while disregarding width and symmetry. Other research experiments with broader and narrower boundaries like documents and sentences as well as fixed width boundaries. Most window methods consider the context as bag of words while some encode relative position of the feature regarding the lexeme in the co-ocurence vectors as well.

If syntactic relations are taken as features, the relation type between the token and lexeme as well as the token are encoded into the co-occurence vector.

1.2 Measures of association with context

Measures of association with context are used to compute values that are included in the co-occurrence vectors. These vales are based on frequencies extracted from corpora.

Before introducing the measures of interest, methods for estimating values from corpora have to be defined. The probability of a feature is computed by maximum likelihood estimate as

$$P(F = f) = \frac{count(F = f)}{count(F)} \quad (1)$$

where the random variable F describes the features distribution.

Furthermore, the conditional probability of a feature given a lexeme is computed as

$$P(F = f|L = l) = \frac{count(F = f, L = l)}{count(L = l)} \quad (2)$$

where the random variable L describes the lemmata distribution.

The joint probability of a feature and a lexeme is computed as

$$P(F = f, L = l) = \frac{count(F = f, L = l)}{count(L)} \quad (3)$$

This expression is derived from the expression that links joint and conditional probability

$$P(A, B) = P(B|A)P(A) = P(A|B)P(B) \quad (4)$$

It has to be stressed here that the estimate of the joint probability can be computed with $count(L)$ or $count(F)$ in the denominator which will result in different values if $V1 \neq V2$.

The most obvious measure of association with context is the plain frequency of co-occurence of a lexeme and a feature.

$$assoc_{freq}(l, f) = count(l, f) \quad (5)$$

This measure has several drawbacks that will be discussed while introducing other measures.

The second most obvious measure would be the relative frequency of a lexeme given a feature, ie. the conditional probability of a feature given a lexeme, ie. a normalized vector which is given by

$$assoc_{prob}(l, f) = P(f|l) \quad (6)$$

Main advantage of this measure in comparison to the absolute frequency is that vectors are set to equal length which makes vectors of very frequent lexeme easier comparable to vectors of infrequent lexemes.

The next possible measure is pointwise mutual information [1] which computes how

often a lexeme and a feature co-occur, compared with what would be expected if they were independent. This measure is computed as

$$assoc_{PMI}(l, f) = \log_2 \frac{P(l, f)}{P(l)P(f)} \quad (7)$$

Main advantage of this measure comparing to the probability measure is that it penalizes co-occurence with features not specific for the lexeme of interest.

Another measure that attempts to capture the same intuition as pointwise mutual information is the t-test statistic which computes the difference between observed and expected means, normalized by the variance. The higher the value of t, the more likely we can reject the null hypothesis that the observed and expected means are the same. This measure is computed in [2] as follows:

$$assoc_{t-test}(l, f) = \frac{P(l, f) - P(l)P(f)}{\sqrt{P(l)P(f)}} \quad (8)$$

1.3 Measures of vector similarity

Measures of vector similarity are used to compare two vectors, ie. lexemes built from association measures for features used to describe the lexemes.

The simplest two measures of vector similarity are the Manhattan and Euclidean distance. The first one calculates the distance between vectors on all dimensions whilst Euclidean distance measures the geometric distance between the two vectors. They are computed by the next two expressions:

$$dist_{manhattan}(\vec{l_1}, \vec{l_2}) = \sum_{i=1}^{N} |l_{1i} - l_{2i}| \quad (9)$$

$$dist_{euclidean}(\vec{l_1}, \vec{l_2}) = \sqrt{\sum_{i=1}^{N} (l_{1i} - l_{2i})^2} \quad (10)$$

After [6], the main problem especially with the Euclidean distance is that it turns out to be very sensible to extreme values, ie. outliers in the vectors. Our intuition is that this holds especially if the vectors are not normalized underestimating the similarity of frequent and infrequent lexemes.

A measure used in information retrieval is the dot product operator from linear algebra. If the vectors are normalized, that measure is equal to the cosine between the two vectors. The cosine similarity measure is computed by

$$sim_{cosine}(\vec{l_1}, \vec{l_2}) = \frac{\sum_{i=1}^{N} l_{1i} * l_{2i}}{\sqrt{\sum_{i=1}^{N} l_{1i}^2} \sqrt{\sum_{i=1}^{N} l_{2i}^2}} \quad (11)$$

The Jaccard measure is also derived from information retrieval. The measure was originally designed for binary vectors. It divides the number of equal features with the number of features in general.

$$sim_{Jaccard\_bin}(l_1, l_2) = \frac{|l_1 \cap l_2|}{|l_1 \cup l_2|} \quad (12)$$

The measure was extended by [4] to vectors with weighted associations as follows:

$$sim_{Jaccard}(\vec{l_1}, \vec{l_2}) = \frac{\sum_{i=1}^{N} min(l_{1i}, l_{2i})}{\sum_{i=1}^{N} max(l_{1i}, l_{2i})} \quad (13)$$

The Dice measure is very similar to the Jaccard measure and is also introduced from information retrieval. It is computed as

$$sim_{Dice\_bin}(l_1, l_2) = \frac{2 * |l_1 \cap l_2|}{|l_1| + |l_2|} \quad (14)$$

There are also equivalents of the Dice measure for weighted associations. The one suggested in [2] is

$$sim_{Dice}(\vec{l_1}, \vec{l_2}) = \frac{2 * \sum_{i=1}^{N} min(l_{1i}, l_{2i})}{\sum_{i=1}^{N} (l_{1i}, l_{2i})} \quad (15)$$

The last measure used in this research is from the family of information-theoretic distributional similarity measures [3]. The intuition of these methods is that two vectors

Table 1: Vjesnik corpus data

number of tokens	79,566,904
number of sentences	3,730,729
number of paragraphs	1,300,785
number of articles	205,686

$\vec{l_1}$ and $\vec{l_2}$ are similar to the extent that their probability distributions $P(f|l_1)$ and $P(f|l_2)$ are similar. The basis of comparing two probability distributions is set by the Kullback-Leibler divergence [5]

$$D(P||Q) = \sum_x P(x) \log \frac{P(x)}{Q(X)} \tag{16}$$

which has the negative property that it is undefined when $Q(x) = 0$ and $P(x) \neq 0$ which is quite often because of the sparseness of the co-occurence matrix.

Therefore there are some alternatives like the Jensen-Shannon divergence [6] which bypasses the negative properties of the Kullback-Leilbler divergence:

$$sim_{JS}(\vec{l_1}, \vec{l_2}) = D(\vec{l_1}||\frac{\vec{l_1} + \vec{l_2}}{2}) + D(\vec{l_2}||\frac{\vec{l_1} + \vec{l_2}}{2}) \tag{17}$$

2 The experiment

In this research the Vjesnik corpus is used. It consists of articles from the on-line version of the Croatian daily newspaper Vjesnik from 1999 to 2007 [8]. The corpus is POS-tagged with the tagger described in [9]. Some data about the corpus is showed in Table 1.

The co-occurence matrix is built for 1,000 most frequent common nouns omitting the first 100. The 1,000 co-occurence vectors are built from co-occurence with common nouns inside a paragraph. Therefore, $V1$ contains 1,000 elements while $V2$ contains all the 15,978 common nouns in the corpus. The highest frequency of a considered lexeme is 26,845 while the minimum frequency is 3,331. The maximum frequency of a feature is 316,911 while the minimum is, as expected, 1.

Out of 1000 considered lexemes five are chosen randomly for the experiment. They are: "ustav" (constitution), "istup" (offset (action)), "suđenje" (trial), "serija" (series) and "prihod" (income).

Four different measures of association with context introduced in the previous section are applied: raw frequency, maximum likelihood estimate, pointwise mutual information and t-test.

Eight different measures of vector similarity introduced in the previous section are applied: L1 (Manhattan distance), L2 (Euclidean distance), cosine similarity, binary Jaccard similarity, Jaccard similarity, binary dice similarity, dice similarity and Jensen-Shannon divergence.

Since binary Jaccard and binary dice similarity work with binary vectors which makes the measures of association with context obsolete, $2 + 6 * 4 = 26$ different experiments for every lexeme are undertaken.

For evaluation a gold standard defined by three human annotators is used. Human annotators are given lists of lexemes possibly related to the selected lexemes. These lists are defined as an union of top twenty answers of all 26 methods. For all five lexemes 390 lexemes (78 on average) are chosen. Each human annotator is given a text file with the lexemes of interest followed by the candidates in alphabetical order. The human annotators are given instructions to grade every lexeme with grades from 1 to 4 where the grades stand for "not similar", "rather not similar", "rather similar" and "similar".

The interannotator agreement between two annotators is calculated as

$$IAA(\vec{g_1}, \vec{g_2}) = 1 - \frac{\sum_{i=1}^{N} |g_{1i} - g_{2i}|}{\sum_{i=1}^{N} 3} \tag{18}$$

The gold standard used to evaluate each of the 26 methods consists of selected lexemes with pairs of semantically similar lexemes and their grades attached. The grades are calculated as the mean of the grades given by human annotators.

In evaluating the 26 methods a loss function based on the inverse ranking is used. It

Table 2: Results of annotation by human annotators

	FREQ	NOC	AG	IAA
ustav	3,299	89	1.96	$.7803 \pm .018$
istup	3,454	103	2.16	$.7584 \pm .024$
suđenje	11,173	51	2.71	$.8519 \pm .019$
serija	5,865	94	1.74	$.7849 \pm .018$
prihod	11,531	53	2.42	$.7862 \pm .022$

uses the ranking vector of the lexeme provided by the method and the grade vector built from the gold standard.

$$L(\vec{r}, \vec{g}) = \sum_{i=1}^{N} \frac{4}{i} - \sum_{i=1}^{N} \frac{g_i}{r_i} \qquad (19)$$

The loss function is computed as the difference between the maximum value (all lexemes receiving the highest score) and the value provided by the gold standard. Each of these two values are computed as fractions of the grade and the ranking, rewarding mostly high grades on high ranks.

3 Results

Table 2 shows the results of the human annotation.

The frequencies of the lexemes are given in Table 2 in column FREQ.

The number of candidates given to human annotators is given in Table 2 in column NOC (number of candidates).

The average grade given to the candidates by the human annotators is 2.2. The average grade regarding a specific lexeme is given in Table 2 in column AG (average grade).

The interannotator agreement is on average $0.7885 \pm 0,0134SE$. The interannotator agreement for a specific lexeme is given in Table 2 in column IAA (interannotator agreement).

The data in Table 2 shows that the most frequent lexemes (*trial* and *income*) have the least number of candidates. In our opinion there are three reasons for that: higher frequency, low level of polysemy and semantic concreteness. We believe that the latter two have greater impact. These two lexemes also have the highest average grade and interannotator agreement.

On the other hand, the lexeme *offset* is most abstract and therefore has the highest number of candidates and the lowest interannotator agreement.

Both *offset* and *series* are highly polysemous and therefore have the highest number of candidates. *Series* has a higher interannotator agreement probably because of the lowest average grade.

The data shows that the highest interannotator agreement have lexemes with best and worst average grades.

The results of the 26 experiments with different measures of vector similarity and as-

Table 3: Loss mean for all lexemes concerning measures used

VS	AWC	loss mean ± SE
js	mle	2.89 ± 0.5618
l1	mle	2.89 ± 0.5779
l2	mle	3.08 ± 0.6104
js	t-test	3.25 ± 0.4459
js	freq	3.63 ± 0.6226
l1	t-test	3.65 ± 0.6404
l1	freq	3.79 ± 0.6332
l2	freq	3.87 ± 0.7309
l1	pmi	3.83 ± 0.7663
l2	pmi	4.10 ± 0.7827
js	pmi	4.10 ± 0.7911
l2	t-test	4.11 ± 0.8062
jaccard	mle	5.35 ± 0.5519
dice	mle	5.35 ± 0.5519
cosine	freq	5.35 ± 1.0453
cosine	mle	5.35 ± 1.0453
cosine	t-test	5.50 ± 1.0313
cosine	pmi	5.65 ± 0.9311
jaccard	t-test	6.31 ± 0.8630
dice	t-test	6.31 ± 0.8630
jaccard	freq	6.34 ± 0.8473
dice	freq	6.34 ± 0.8473
jaccard	pmi	6.60 ± 0.7742
dice	pmi	6.60 ± 0.7742
jaccard_bin		6.89 ± 0.6025
dice_bin		6.89 ± 0.6025

Table 4: Loss mean for specific lexemes

lexeme	loss mean ± SE
ustav	5.43 ± 0.3470
istup	4.56 ± 0.2710
suđenje	3.18 ± 0.3261
serija	7.12 ± 0.3286
prihod	4.32 ± 0.2533

Table 5: Minimum loss and methods applied for specific lexemes

lexeme	VS	AWC	min loss
ustav	js	mle	2.99
istup	l1	mle	1.88
suđenje	js	mle	1.41
serija	l2	mle	4.57
prihod	js	freq	2.57

sociation with context are given in Table 3. They are sorted by the loss mean and the standard error.

The results show that the strongest variable regarding the loss mean is the vector similarity measure. The first half of the list consists of a combination of the Jensen-Shannon divergence and L1 and L2 measures. The second half consists of the remaining vector similarity measures - cosine, Jaccard and Dice measures. The binary measures show highest loss mean.

Best results concerning the measure of association with context shows the maximum likelihood estimate, followed by t-test and raw frequency. Pointwise mutual information shows weakest results.

The reason for the success of the MLE measure is in our opinion twofold - on one hand Jansen-Shannon divergence, the most successful measure, expects this measure and on the other hand only common nouns are used as features which makes additional weighting less important.

In Table 4 and Table 5 loss mean and the minimum for every specific lexeme are showed. As expected, the loss mean variable strongly correlates with the average grade variable and minimum loss variable.

The best result in this research is achieved for the lexeme *trial* with the following top 20 results: trial, witness, indictment, verdict, accused, attorney, judge, prosecutor, custody, investigation, prison, testimony, murder, female judge, evidence, charge, bar, sentence, appeal, crime. All the original Croatian lexemes are rather monosemous and specific for the topic.

4 Conclusion

In this research several methods of vector similarity and association with context are evaluated in the process of extracting semantically similar lexemes from the corpus. In building the co-occuring matrix the sentence window is used. Co-occurence vectors of only 1,000 most frequent common nouns are built only with information about co-occurring common nouns. The gold standard is built by three human annotators using a union of top results of all methods. All together, 26 different combinations of vector similarity and association with context are evaluated.

The results show that Jensen Shannon divergence and L1 and L2 measures outperform the remaining vector similarity measures.

The best measure of association with context is the maximum likelihood estimate. The reason that more sophisticated measures like the t-test or pointwise mutual information underperformed probably lies in the fact that only common noun co-occuring information is used which is generally very informative.

The distribution of grades given by human annotators shows that there is a high amount of medium grades. Namely, the methods extract a large amount of topically similar lexemes that are mostly graded with grades 2 and 3. We claim that the reason for that is the size of the window applied. Our assumption is that a narrower window would produce more real synonyms and near-synonyms. Such a large window has the consequence that nouns that often co-occur are represented by more similar vectors than it would be if the window was narrower.

Further research has to emphasize primarily the variable of window size. Additionally, a larger number of lexemes has to be included

in the matrix and also experimented on.

The decision about taken features has to be experimented with since in this research only co-occurence with common nouns is considered.

Chunking and parsing the corpus should also improve the results since research on English showed that best results are obtained by including syntactic relations into co-occurrence vectors.

References

[1] K. W. Church and P. Hanks. Word association norms, mutual information, and lexicography. In *Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics*, pages 76–83, Vancouver, B.C., 1989. Association for Computational Linguistics.

[2] J. R. Curran. *From Distributional to Semantic Similarity.* PhD thesis, University of Edinburgh, 2004.

[3] I. Dagan, L. Lee, and F. C. N. Pereira. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1-3):43–69, 1999.

[4] G. Grefenstette. *Explorations in Automatic Thesaurus Discovery.* Kluwer Academic Publishers, Norwell, MA, USA, 1994.

[5] S. Kullback and R. A. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.

[6] L. Lee. Measure of distributional similarity. In *Proceedings of ACL'99: the 37th Annual Meeting of the Association for Computational Linguistics*, pages 25–32, Vancouver, B.C., 1999. Association for Computational Linguistics.

[7] H. Schutze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[8] Vjesnik on-line, http://www.vjesnik.hr, 1999-2007.

[9] Ž. Agić and M. Tadić. Evaluating morphosyntactic tagging of croatian texts. In *LREC2006 Proceedings*, Genoa-Paris, 2006. ELRA.