3rd Annual South-East European Doctoral Student Conference

Infusing Research and Knowledge in South-East Europe

edited by
Iraklis Paraskakis
Andrej Luneski

SEERC
SOUTH-EAST EUROPEAN RESEARCH CENTRE

Volume 2

[Nakov, 2008a] Nakov S. "Automatic Acquisition of Synonyms Using the Web as a Corpus". *Proceedings of the 3rd Annual South-East European Doctoral Student Conference (DSC 2008)*, Volume 2, pages 216-229, Thessaloniki, Greece, 2008

Automatic Acquisition of Synonyms Using the Web as a Corpus

Svetlin Nakov[1]

[1]*Sofia University "St. Kliment Ohridski", Department of Mathematics and Informatics, 5 James Baucher Bvld., Sofia, Bulgaria, nakov [at] fmi.uni-sofia.bg*

We present an original algorithm for automatic acquisition of synonyms from text. The algorithm measures the semantic similarity between pairs of words by comparing their local contexts extracted from the Web by series of queries against the Google search engine. The results show 11pt average precision of 63.16%.

Keywords

Automatic synonym acquisition, semantic similarity measure, Web as a corpus, Web mining.

1. Introduction

Synonyms are important for solving various problems in Natural Language Processing (NLP) such as text summarization, question answering, text generation, search query expansion, etc.

In the present paper, we set the objective to design an algorithm for automatic extraction of pairs of synonyms from a text corpus. The results can be used to create linguistic resources, such as general and domain-specific thesauri and lexicons.

We use the Web as a large corpus which can be efficiently searched. Our approach is based on performing series of queries against a Web search engine and analyzing the returned excerpts of texts (snippets) in order to extract contextual semantic information which we use to measure the semantic similarity between pairs of words and thus to approximate synonymy.

It is considered that the *local context* of a given word (few words before and after the target captured word) contains words that are semantically related to it [Hearst, 1991]. Given a pair of words, we extract their local contexts from the snippets returned by the search engine and we measure the semantic similarity between these words by calculating the similarity between their local contexts. Finally, the measured similarity is used to determine whether the words are likely to be synonyms or not.

The algorithm used for measuring semantic similarity is an adaptation of the algorithm for measuring cross-lingual semantic similarity described in [Nakov et al., 2007a].

In the performed experiments we process Russian texts used for teaching students studying fine arts. We chose Russian and fine arts terminology because of the high volume of such texts available in Internet and the great number of full synonyms in this domain. While the algorithm is general enough and should work for other languages, our present experiments are limited to Russian only.

We start with extracting a list of all terms from the text that are interesting from a linguist's point of view. We can also use a subset of them, e.g. nouns only, or all words in the text.

Using a series of queries against Google, we automatically measure the semantic similarity between each pair of words from the list. Our hypothesis is that synonyms should have higher level of semantic similarity compared to nonsynonyms. The results of our experiments show that this expectation is true in most cases.

In this paper, we show that it is possible (with a minimum human intervention) to extract automatically all pairs of synonyms from a list of terms built from a terminological text. We propose few modifications of the algorithms for measuring semantic similarity using the Web and we study how different parameters affect the quality of the results (precision and recall).

2. Method for Automatic Synonyms Extraction

Our algorithm for automatic extraction of synonyms from a list of words is based on measuring the semantic similarity between pairs of words by querying a Web search engine (e.g., Google) and analyzing the returned results. The semantic similarity is a number between 0 and 1 indicating the degree of similarity.

The words used to find synonyms come as a list. It is possible to process all words in the text or some subset of them. For example, in order to avoid unnecessary computations, we can use grammatical glossary to filter out words belonging to different parts of speech.

The algorithm measures semantic similarity between each pair of words from our list. Our hypothesis is that synonyms, being words with very similar meanings, should have higher semantic similarity than pairs of nonsynonyms. If we order all pairs of words by their measured semantic similarity, we can expect to obtain identical words in the beginning, followed by synonyms, followed by partial synonyms, followed by other words which are less similar by meaning (like hypernyms and hyponyms), and finally all words that are entirely different.

Since our semantic similarity measures to what extent two words have a similar meanings, it is possible to get inaccurate results for some pairs of words and incorrectly to classify them as synonyms. For example *абрис* and *контур* are semantically related because both mean the same concept (*contour* in Russian), but at the same time words like *синий* (*blue* in Russian) and *красный* (*red* in Russian) are also semantically related because both are colors. Therefore, extracting synonyms by measuring semantic similarity only is not possible without human intervention, but our experiments show that this intervention could be minimal.

2.1. Semantic Similarity Measured by Contexts

The algorithm for measuring semantic similarity between two words is based on analysis of the local context in which these words appear and follows the idea that words appearing in similar context should have similar meanings. For example the words *художник* (*artist* in Russian) and *картина* (*panting* in Russian) are semantically related since both appear in sentences about artists, painters, painting, pictures, brushes, tints, art galleries, and other terms from the fine arts.

Some sentences can be quite long, and it is not clear what part of them contains the context of the given word. Most linguists consider only the so called *local context* of the given word in a sentence which consists of few words before and after that word. As an example let us examine the word *painting* in the following sentence:

> You will learn watercolor techniques, oil *painting* techniques, chalks and freehand styles of *painting*, guided by Jane who has over 25 years of experience as a professional portrait artist and painter.

The local context (e.g. three words before and after it) of the word *painting* in the above sentence contains the following words: *watercolor*, *techniques*, *oil*, *techniques*, *chalks*, *and*, *freehand*, *styles*, *guided*, *by*, *Jane*. If we take the basic forms (the lemmata) of these words and remove the repeating words and functional words such as prepositions, conjunction and pronouns, we will end up with the following few words that form the local context of the word *painting* in this sentence: *watercolor*, *technique*, *oil*, *chalk*, *freehand*, *style*, *guide*, *Jane*.

painter		painting	
painter	422	painting	461
painting	262	buy	386
paint	202	expensive	345
art	167	famous	205
gallery	94	gallery	183
famous	84	big	176
buy	72	art	188
big	56	painter	98
expensive	3	paint	91
camera	0	camera	2

Table 1. Frequency vectors for the terms "painter" and "painting".

Most of these words are semantically related to *painting*, but some of them are not. If we take the word *painting* and a sufficiently large number of sentences containing that word (e.g., 1,000) and we extract from them all the words from its local context, we could expect that the most frequently appearing words to be semantically related to *painting*. These words should contain terms from fine arts and painting such as *painter*, *paint*, *brush*, *art*, *artist*, *technique*, and *style*. Accidentally found words like *guide* and *Jane* should appear quite rarely if we take a sufficiently large set of arbitrary sentences.

Now let us take two words and extract the frequently appearing words in their local contexts taken from sufficiently large number of sentences. If these two words are semantically similar we could expect their context words and respective numbers of occurrences to be also similar.

We can formalize the above ideas by assigning frequency vectors formed out of the words in the local contexts of the target words and measure the similarity between these vectors. For example for the words *painter* and *painting*, we could have the frequency vectors of the words in their contexts as shown in table 1 (with abridgments).

As dimensions of the vectors we take all words appearing in the contexts of at least one of the words and as coordinates we take their frequencies. For words not appearing in given context, we assume frequency 0. Therefore, we obtain the frequency vectors (with abridgments) shown in Table 2.

word	**vector 1 (painter)**	**vector 2 (painting)**
painter	422	98
painting	262	461
paint	202	91
art	167	188
gallery	94	183
famous	84	205
buy	72	386
big	56	176
expensive	3	345
camera	0	2

Table 2. Comparison of the frequency vectors for the terms "painter" and "painting".

We compute the similarity between the vectors as the cosine in the n-dimensional Euclidean space. Thus we obtain a number between 0 and 1, which is a numerical measure for the semantic similarity between two words (higher value means more similar words).

2.2. Semantic Similarity Measured by Web Contexts

The World Wide Web (WWW) contains the largest set of text corpora in the world in a number of languages (including Russian) and provides efficient searching capabilities through the Web search engines. This motivates us to use the Web as a source of local context information for measuring semantic similarity between pair of words. We will describe a method for extraction of local context from the Web (*web context*), similar to the one described in [Nakov et al., 2007a].

For the extraction of the local context for a given word from the Web, we use a query against a Web search engine in which we request 100 results in the target language (in our experiments Russian). Using a sequence of 10 such queries, we can obtain up to

1,000 query results (Google sets explicit limits to never return more than 1,000 results). Each result contains a title and an excerpt (snippet) of text containing the word we searched for. For example, if we search for *painting* in English, we could get the following list of titles and text snippets, as shown in Table 3.

Painting - Wikipedia, the free encyclopedia
Painting, meant literally, is the practice of applying color to a surface (support) such as, e.g. paper, canvas, wood, glass, lacquer or concrete. ...
Painting - Exterior & Interior House Painters - Faux Finishing ...
Painting information, articles, pictures, **painting** ideas & more. Free price quotes from local exterior & interior **painting** contractors.
About.com **Painting** -- How-To Articles, **Painting** Tips, Projects ...
Whether you're into **painting** with oils, acrylics, watercolors, pastels, or mixed media, here you'll find essential how-to information, tips, ...
...

Table 3. List of titles and text snippets returned by Google for the word "painting".

In the titles and snippets returned by the search engine, we first convert all letters to lowercase and we extract all words.

We then remove all functional words (prepositions, pronouns, conjunctions, particles, interjections, and some adverbs) as well as all words with less than 3 letters. Such words do not bring semantic information about the searched word and should be omitted because they only distort the results.

Then we go through the extracted words sequences and when we find the target word or one of its forms, we take 3 words before and after it (the number 3 here we call *context size*). We consider these words part of the Web context.

We apply lemmatization (replace all words with their basic form), e.g. replace *paintings* with *painting*. For this purpose, we use a rich grammatical dictionary of Russian.

Now we have all words which appear in the local Web context of the target word and their corresponding frequencies (frequency vectors).

We measure the semantic similarity between two words by calculating the cosine between the frequency vectors of these words taken from their Web contexts.

2.3. TF.IDF Weighting

In information retrieval, TF.IDF weighting is a common technique for improving the search quality. The number TF.IDF (term frequency times inverted document frequency) is a statistical measure that shows how important is a certain word for a given

document in a set of documents. The importance of the word increases proportionally to the number of its occurrences in the document but decreases proportionally to the total number of documents containing it. It was shown that if words' frequency is weighted according to their importance, the search quality improves [Sparck-Jones, 1972].

To apply TF.IDF weighting in our semantic similarity measure algorithm, we do the following: when we get the first 1000 query results from the Web search engine for a given word w, we directly compute the frequencies $TF[w_i]$ of all words w_i in its context by dividing the occurrences of w_i to the total number of words in the context of w (including duplicates). After that, we compute $IDF[w_i]$ by dividing the total number of documents indexed by Google (we assume they are about eight billions) to the number of occurrences in Google of w_i. Finally, we take $log_2(IDF[w_i])$ and multiply it by $TF[w_i]$ and thus we compute the weighted frequency of the word w_i in the frequency vector of w. The obtained weighted frequency vector we use for more measuring semantic similarity more accurately.

2.4. Semantic Similarity Measured through Reverse Context

When we extract the Web context for a given word, often semantically unrelated words fall in. For example, Internet terminology like *site*, *page*, *blog*, *online*, *forum*, *web*, *network*, *home*, *link*, *menu*, *message*, *download*, etc. are likely to appear in the context of almost any word, despite not being semantically related to it. Removing such words from the context is expected to improve accuracy when measuring semantic similarity with Web contexts [Nakov et al., 2007b].

The *reverse context lookup* technique is based on the idea that if two words are semantically related, the first one should often appear in the context of the second one, and at the same time, the second one should also often appear in the context of the first one.

For example in the context of the word *painting*, words like *painter*, *gallery* and *art* appear often, but so do parasite words like *order*, *news* and *site* as well. If we search the Web for the first three words, we shall convince ourselves that *painting* appears often in their contexts. However, if we search for the last three words, we will find that in their contexts *painting* almost does not appear.

We can formalize this idea as follows. Let $F(x,y)$ be number of appearances of y in the Web context of x. Consider some word w and all the words w_i from its Web context along with their frequencies $F(w,w_i)$. Now let us extract from the Web for each word w_i the number of reverse occurrences $F(w_i,w)$ of the word w in the context of w_i (*reverse context*). Finally, we can obtain a vector of the co-occurrences of the word w with all words from its context. It consists of all words w_i with frequencies:

$$min(F(w, w_i), F(w_i, w))$$

The obtained frequency vector contains more accurate semantic information than the simple frequency vector because for each word it holds the minimal number of co-occurrences of the word with each word from its context.

When computing the co-occurrence frequency vector we can ignore words that occur in the co-occurrence frequency vector infrequently (e.g., three times or less) because this could have happened by chance. By modifying this parameter (frequency threshold), we can affect the accuracy of the results.

2.5. Synonyms Extraction by Measuring Semantic Similarity

Our method for extraction of synonyms by measuring semantic similarity is based on the hypothesis that synonym pairs should have higher semantic similarity compared to nonsynonyms.

If we are given a set of words and we measure the semantic similarity between each two of them, after sorting the pairs of words in a list in decreasing order by their semantic similarity, we can expect that synonyms are at the beginning of the list, followed by other semantically similar words, followed by words that are unrelated.

3. Experiments and Results

The experiments we preformed focus on studying and analyzing our algorithms for measuring semantic similarity extracted from the Web and their usage for the automatic discovery of synonyms. We performed experiments without and with using the reverse context and TF.IDF weighting and with various thresholds for the minimal frequency of the words in the context.

3.1. Resources Used

For the purposes of our experiments and for the implementation of our algorithms for measuring semantic similarity using the Web, we used the following resources:

- **Online Web search engine Google**[1]. We performed queries for 82,645 Russian words and collected the first 1,000 results for each of them.
- **Grammatical dictionary of the Russian language**, created in the Linguistic Modeling Laboratory, Institute for Parallel Processing, Bulgarian Academy of Sciences [Paskaleva,2002]. The dictionary contains about 1,500,000 wordforms and about 100,000 lemmata. Each dictionary entry contains wordform, corresponding lemma, followed by morphological and grammatical information.
- **List of the functional (stop) words in Russian**. Contains 507 words (prepositions, pronouns, conjunctions, particles, interjections and some adverbs). Created manually.

Our algorithm is general and can be applied to many languages. It does not require resources that are hard to find. The only resource that is not publicly available for any language is the grammatical dictionary. It is good to have it for highly inflectional languages like Russian, but this is less important for languages like English.

3.2. Test Data Set

In the experiments that we preformed, we used a list of 94 words from the Russian fine arts terminology, prepared manually by a linguist based on a set of study texts for students of fine arts. We selected only terms that occur in Google at least 5,000 times in order to have a statistical precision. Terms that occur in too small number of pages on

[1] http://www.google.com

the Web (e.g., just 5 times) cannot be analyzed statistically because the extracted context will be too small and not enough meaningful.

Below is an excerpt from our list of 94 words:

> *абрис, адгезия, алтарь, амулет, асфальт, вохрение, выжигание, гематит, диамант, жезл, закрепление, ...*

There are 50 pairs of synonyms among these words, which we expect to be found by our algorithms.

3.3. Experiments

In all experiments our selected 94 words (terms from fine arts terminology) are processed in pairs and for each of them the semantic similarity is calculated. As a result, we obtained a list of 4,371 word pairs ordered in descending order by their similarity.

We measure the accuracy by *precision* and *recall*, which come from information retrieval. We experimented with few variations of the algorithm for measuring semantic similarity:

- **RAND** – returns a random ordering of all the pairs of words. We use this as a base for comparison with the other algorithms.
- **SIM** – the major algorithm for extraction of semantic similarity from the Web (described in detail in 2.2) with context size of 3 words, without analyzing the reverse context, with lemmatization.
- **SIM+TFIDF** – modification of the SIM algorithm with TD.IDF weighting (described in detail in 2.3).
- **REV2**, **REV3**, **REV4**, **REV5**, **REV6**, **REV7** – modifications of the SIM algorithm using the "reverse context lookup" technique (described in detail in 2.4) with the following frequency thresholds for the context words: 2, 3, 4, 5, 6 and 7.

3.4. Results

There are few well-known metrics for evaluation of information retrieval algorithms:

- **Precision @ n** – specifies what portion of the first *n* results are correct.
- **Recall @ n** – specifies what portion of all correct results are in the set of the first *n*.

The precision and recall are numbers between 0 and 1, and are typically expressed in percentages.

In our case, the algorithms for synonyms extraction using the Web as a corpus return a list of pairs of words and some of them are synonyms while other are not. We compute *precision @ n* by dividing the number of synonyms in the first *n* pairs by the number *n*. We compute *recall @ n* by dividing the total number of synonyms that exist in the data set (50) by the number of synonyms in the first *n* pairs. In practice, to evaluate a given

algorithm, we need to know only how many synonyms are found in the first *n* pairs of words in the list. The more the words are, the better the algorithm is.

Table 4 shows an excerpt of the results obtained using the SIM algorithm and their corresponding precision and recall.

n	Word 1	Words 2	Semantic Similarity	Syno-nyms	Precision @ n	Recall @ n
1	выжигание	пирография	0.433805	yes	100.00%	2%
2	тонирование	тонировка	0.382357	yes	100.00%	4%
3	гематит	кровавик	0.325138	yes	100.00%	6%
4	подрамок	подрамник	0.271659	yes	100.00%	8%
5	оливин	перидот	0.252256	yes	100.00%	10%
6	полирование	шлифование	0.220559	no	83.33%	10%
7	полировка	шлифовка	0.216347	no	71.43%	10%
8	амулет	талисман	0.200595	yes	75.00%	12%
9	пластификаторы	мягчители	0.170770	yes	77.78%	14%
10	родонит	орлец	0.168245	yes	80.00%	16%

Table 4. Precision and recall obtained by the SIM algorithm.

The results of all evaluated algorithms are given in Table 5.

Algorithm	1	5	10	20	30	40	50	100	200	Max
RAND	0	0.1	0.1	0.2	0.3	0.4	0.6	1.1	2.3	**50**
SIM	**1**	**5**	**8**	15	18	23	25	39	**48**	**50**
SIM+TFIDF	1	4	**8**	**16**	**22**	27	29	**43**	**48**	**50**
REV2	1	4	**8**	**16**	21	27	32	42	43	46
REV3	1	4	**8**	**16**	20	**28**	32	41	42	46
REV4	1	4	**8**	15	20	**28**	**33**	41	42	45
REV5	1	4	**8**	15	20	**28**	**33**	40	41	42
REV6	1	4	**8**	15	**22**	**28**	32	39	40	42
REV7	1	4	**8**	15	21	27	30	37	39	40

Table 5. Comparison of the algorithms (number of synonyms in the results).

Instead of precision and recall, in table 5, we give the number of synonyms found in the first 1, 5, 10, 20, 30, 40, 50, 100 and 200 results. The best values are given in bold. The last column shows the total number of synonyms found by the corresponding algorithm. This number does not always reach the maximal value of 50 because most of the algorithms return no semantic similarity (value of 0) for large amount of the pairs from the test data set and thus we cannot assign certain positions in the ordered list for them to be able to evaluate the accuracy.

We evaluated the SIM algorithm also using ”11-pt average precision“, a widely used metric in information retrieval which combines precision and recall in a single number

[Salton, 1983]. 11-point average precision is computed by averaging the values in 11 points respectively for recall of 0%, 10%, 20%, … and 100%. The obtained result for SIM algorithm is 58.98%, and for SIM+TFIDF algorithm is 63.16%. For the other algorithms 11pt average precision is not defined (their recall never reaches 100%).

4. Discussion

In table 5, we can see that the proposed algorithms arrange most of the synonyms at the beginning of the produced ordered lists of pairs. The improvement over the random ordering (RAND) is huge, but the algorithms are still not perfect. Below we compare the algorithms in more detail and we discuss what causes the errors.

4.1. Comparison of the Algorithms

Figure 1 shows the precision/recall curves for the algorithms RAND, SIM, SIM+TFIDF and REV4.

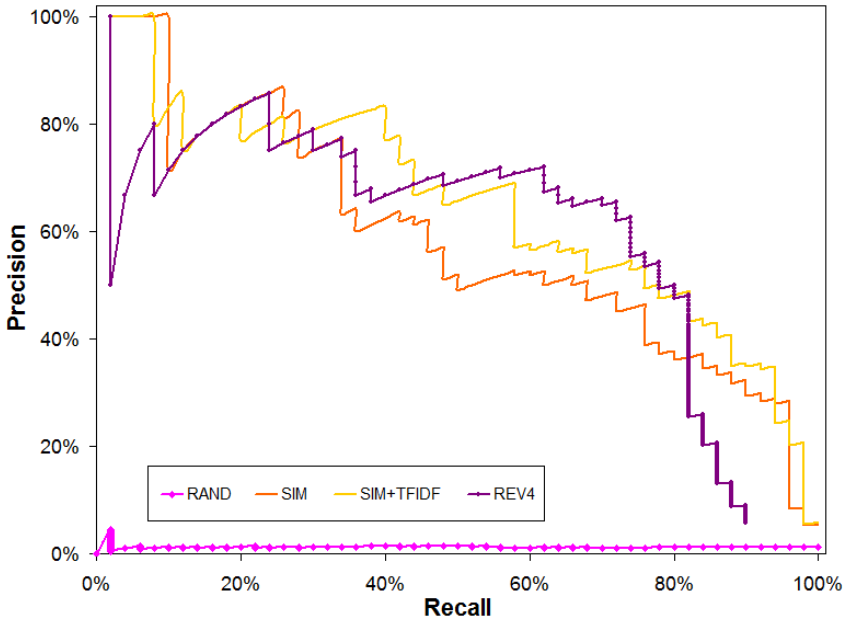

Figure 1. Precision / recall curves for the algorithms RAND, SIM, SIM+TFIDF and REV4.

The major SIM algorithm starts well with 5 correct synonyms and precision @ 5 of 100%. For the top 10 ranked pairs and in the first 20 pairs the precision remains very high: 80%. For the top 40 pairs, the algorithm lists 56% of all the synonyms and its precision is still 70%. For the top 100 pairs, the precision falls to 40%, but the recall is over 80%. For the top 200 pairs, almost all synonyms are listed (recall 96%), but the precision drops to 24%. The SIM algorithm lists almost all synonyms in the first 100 results (which are only 2.11% of all 4,371 pairs in the list).

Applying TF.IDF weighting improves the accuracy of the SIM algorithm and yields better precision and recall. Respectively, the graphic of SIM+TFIDF is located above the graphic of the SIM algorithm most of the time.

Applying the *reverse context lookup* technique improves the SIM algorithm by increasing its precision for the top 40-50 pairs in the list (algorithms REV2 – REV7), but overall decreases the recall. The reverse context lookup technique works better than TF.IDF weighting for the beginning of the list (top 30-50 pairs), but is worse thereafter.

Modifying the threshold of the frequency used in the reverse context lookup has little impact on the accuracy. Lower threshold causes a lower precision at the beginning of the list and an overall higher recall. Higher thresholds improve the precision for the top 40-50 pairs, but lowers the overall recall.

4.2. Known Problems

The accuracy of the obtained results for all algorithms is less than 100%. Below are the most important reasons for that:

- There is an inaccuracy when measuring the semantic similarity by our algorithms because not always words appearing in similar contexts are similar in meaning.
- Using the Web as a corpus limits the extraction of the local contexts to the first 1,000 results only. Since commercial and news sites are typically ranked higher by Google, the top 1,000 results are not a representative sample of all texts on the Web.
- High semantic similarity is typical for synonym pairs, but it is not limited to synonyms only. Our algorithms assume that words appearing in similar contexts are similar, but this does not directly mean that they are synonyms. For example, the colors *blue* and *red* are semantically related (because both mean a kind of color), but are not synonyms. This causes significant errors during synonyms extraction and can be seen from the obtained results. Fixing this problem would be the most important challenge in our future work.

5. Related Work

Most of the automatic synonym extraction methods are based on distributional hypothesis, that semantically related words appear in close contexts [Harris, 1954]. This hypothesis provides a key point for many other synonyms retrieval algorithms: *contexts retrieval and comparison*. In its essence, our method is also based on context retrieval and comparison, but we use the Web as a corpus for measuring semantic similarity and in this way we do not depend on other linguistic resources (e.g., large text corpora).

Algorithms, based on the distributional hypothesis, are proposed by [Lin, 1998] and [Curran et al., 2002]. In these papers, the contexts are defined based on predefined grammatical relations that are retrieved from a language corpus. They also take into account the similarity between the retrieved contexts.

The main problem of all the above methods is the difficulty to distinguish synonyms from other semantically similar pairs of words such as hyponyms, hypernyms, antonyms, etc. We expect that synonyms, being more strongly related, will have higher degree of similarity than e.g. hyponyms or hypernyms, but this is not always the case. The problem persists in our work as well.

The problem is partially solved by [Plas et al., 2006], who use two parallel corpora aligned at the word level using GIZA++ [Och, 2003], from which the corresponding sentences and all probable translations between word pairs in both languages are retrieved. As a context for a given word in the first language, the set of all its probable translations in the other language are used. Then the semantic similarity between the two words is measured as a smilarity between their contexts. This approach allows for a more accurate distinction between synonyms and other semantically related words, because antonyms and hypernyms rarely get aligned. The disadvantage of this approach is that it requires a big parallel corpus, which can be unavailable. It will also not work for uncommon words, which are almost not met in the corpus.

[Hagiwara et al., 2007] propose to measure semantic similarity using local contexts extended with indirect retrieval of additional contextual words. In particular, after the local context C for a given word has been retrieved, the words from the local contexts of all words in C are added to the local context of that target word as well. In this way, the semantic information is enriched and thus the accuracy of measuring semantic similarity is improved. The only disadvantage is that this approach of retrieving context from the Web is too expensive because of the high number of search queries needed to retrieve the indirect context words.

The idea of using the Web as a corpus has been used by many scientists solving different problems (see [Kilgarriff et al., 2003] for an overview). Some of them use Web search engines for finding how many times a word or phrase is met and calculating pointwise mutual information [Inkpen, 2007], whereas others directly retrieve context from text snippets returned by Web search engines [Nakov et al., 2007b].

The idea of retrieving information from text snippets returned by a Web search engine is used in [Chen et al., 2006]. The model they introduce is based on the idea that if two words X and Y are semantically bound, then searching for X should cause Y to appear often in the results, and vice versa: searching for Y should cause X to appear often in the results. In this approach, context words are completely ignored (except for X and Y) and their semantics are not used. As it is later discovered [Bollegala et al., 2007], this produces incorrect zero semantic similarity for most of the processed pairs.

[Sahami et al., 2006] use the Web as a corpus to measure the semantic similarity between pairs of short text fragments (search requests), thus gaining automatic requests expansion and offering alternative requests. For this purpose, they retrieve the contexts of the pairs of short texts from the content of the documents returned after searching, and they then compare the most frequent words from these documents. In contrast, we do not compare the content of the documents but only the snippets returned by a Web search engine, which requires much less resources and yields better results since not all words from the document are taken into account but only the ones in the local context.

[Bollegala et al., 2007] combine retrieval of information about the number of occurrences of two words (both together and individually) from a Web search engine, with

retrieval of information from text snippets returned by the search engine. They automatically discover lexico-syntactic templates for semantically related and unrelated words using WordNet, and they train a support vector machine (SVM) classifier. The learned templates are used for extracting information from the text fragments returned by the search engine. Finally, the results are combined. The method is more complicated than the one we propose and requires extra resources for training the SVM.

An interesting approach for finding synonyms and lexicalizations from the Web is described in [Sanchez et al., 2005]. They start with a taxonomy of terms relevant to a specific domain built automatically for a given keyword based on series of searches in Google. They then search the Web for the longest multiword terms extracted from the taxonomy after removing the target keyword and assume that synonyms should be found on the same position where the original keyword was. The approach is quite original, but addresses a different problem: find possible synonyms for a given word.

A major advantage of our method is that it does not require large corpora or other resources like WordNet, which are not available for some languages.

6. References

[1] Hearst M. (1991). "Noun Homograph Disambiguation Using Local Context in Large Text Corpora". In *Proceedings of the 7th Annual Conference of the University of Waterloo Centre for the New OED and Text Research*, Oxford, England, pages 1-22.

[2] Nakov P., Nakov S., Paskaleva E. (2007a). “Improved Word Alignments Using the Web as a Corpus”. In *Proceedings of RANLP'2007*, pages 400-405, Borovetz, Bulgaria.

[3] Nakov S., Nakov P., Paskaleva E. (2007b). “Cognate or False Friend? Ask the Web!”. In *Proceedings of the Workshop on Acquisition and Management of Multilingual Lexicons, held in conjunction with RANLP'2007*, pages 55-62, Borovetz, Bulgaria.

[4] Sparck-Jones K. (1972). “A Statistical Interpretation of Term Specificity and its Application in Retrieval”. *Journal of Documentation*, volume 28, pages 11-21.

[5] Salton G., McGill M. (1983), *Introduction to Modern Information Retrieval*, McGraw-Hill, New York.

[6] Paskaleva E. (2002). “Processing Bulgarian and Russian Resources in Unified Format”. In *Proceedings of the 8th International Scientific Symposium MAPRIAL*, Veliko Tarnovo, Bulgaria, pages 185-194.

[7] Harris, Z. (1954). "Distributional structure”. *Word*, 10, pages 146-162.

[8] Lin D. (1998). "Automatic Retrieval and Clustering of Similar Words". In *Proceedings of COLING-ACL'98*, Montreal, Canada, pages 768-774.

[9] Curran J., Moens M. (2002). "Improvements in Automatic Thesaurus Extraction". In *Proceedings of the Workshop on Unsupervised Lexical Acquisition, SIGLEX 2002*, Philadelphia, USA, pages 59-67.

[10] Plas L., Tiedemann J. (2006). "Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity". In *Proceedings of COLING/ACL 2006*, Sydney, Australia.

[11] Och F., Ney H. (2003). "A Systematic Comparison of Various Statistical Alignment Models". *Computational Linguistics*, 29 (1), 2003.

[12] Hagiwara M., Ogawa Y., Toyama K. (2007). "Effectiveness of Indirect Dependency for Automatic Synonym Acquisition". In *Proceedings of CoSMo 2007 Workshop, held in conjuction with CONTEXT 2007*, Roskilde, Denmark.

[13] Kilgarriff A., Grefenstette G. (2003). "Introduction to the Special Issue on the Web as Corpus", *Computational Linguistics*, 29(3):333–347.

[14] Inkpen D. (2007). "Near-synonym Choice in an Intelligent Thesaurus". In *Proceedings of the NAACL-HLT*, New York, USA.

[15] Chen H., Lin M., Wei Y. (2006). "Novel Association Measures Using Web Search with Double Checking". In *Proceedings of the COLING/ACL 2006*, Sydney, Australia, pages 1009-1016.

[16] Sahami M., Heilman T. (2006). "A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets". In *Proceedings of 15th International World Wide Web Conference*, Edinburgh, Scotland.

[17] Bollegala D., Matsuo Y., Ishizuka M. (2007). "Measuring Semantic Similarity between Words Using Web Search Engines", In Proceedings of the 16th International World Wide Web Conference (WWW2007), Banff, Canada, pages 757-766.

[18] Sanchez D., Moreno A. (2005), "Automatic Discovery of Synonyms and Lexicalizations from the Web". Artificial Intelligence Research and Development, Volume 131, 2005.