A Semantic Similarity Measure Based on Lexico-Syntactic Patterns

Alexander Panchenko, Olga Morozova and Hubert Naets
Center for Natural Language Processing (CENTAL)
Université catholique de Louvain – Belgium
{Firstname.Lastname}@uclouvain.be

Abstract

This paper presents a novel semantic similarity measure based on lexico-syntactic patterns such as those proposed by Hearst (1992). The measure achieves a correlation with human judgements up to 0.739. Additionally, we evaluate it on the tasks of semantic relation ranking and extraction. Our results show that the measure provides results comparable to the baselines without the need for any fine-grained semantic resource such as WordNet.

1 Introduction

Semantic similarity measures are valuable for various NLP applications, such as relation extraction, query expansion, and short text similarity. Three well-established approaches to semantic similarity are based on WordNet (Miller, 1995), dictionaries and corpora. WordNet-based measures such as *WuPalmer* (1994), *LeacockChodorow* (1998) and *Resnik* (1995) achieve high precision, but suffer from limited coverage. Dictionary-based methods such as *ExtendedLesk* (Banerjee and Pedersen, 2003), *GlossVectors* (Patwardhan and Pedersen, 2006) and *WiktionaryOverlap* (Zesch et al., 2008) have just about the same properties as they rely on manually-crafted semantic resources. Corpus-based measures such as *ContextWindow* (Van de Cruys, 2010), *SyntacticContext* (Lin, 1998) or *LSA* (Landauer et al., 1998) provide decent recall as they can derive similarity scores directly from a corpus. However, these methods suffer from lower precision as most of them rely on a simple representation based on the vector space model. *WikiRelate* (Strube and Ponzetto, 2006) relies on texts and/or categories of Wikipedia to achieve a good lexical coverage.

To overcome coverage issues of the resource-based techniques while maintaining their precision, we adapt an approach to semantic similarity, based on lexico-syntactic patterns. Bollegala et al. (2007) proposed to compute semantic similarity with automatically harvested patterns. In our approach, we rather rely on explicit relation extraction rules such as those proposed by Hearst (1992).

Contributions of the paper are two-fold. First, we present a novel corpus-based semantic similarity (relatedness) measure *PatternSim* based on lexico-syntactic patterns. The measure performs comparably to the baseline measures, but requires no semantic resources such as WordNet or dictionaries. Second, we release an Open Source implementation of the proposed approach.

2 Lexico-Syntactic Patterns

We extended a set of the 6 classical Hearst (1992) patterns (1-6) with 12 further patterns (7-18), which aim at extracting hypernymic and synonymic relations. The patterns are encoded in finite-state transducers (FSTs) with the help of the corpus processing tool UNITEX [1]:

1. such NP as NP, NP[,] and/or NP;
2. NP such as NP, NP[,] and/or NP;
3. NP, NP [,] or other NP;
4. NP, NP [,] and other NP;
5. NP, including NP, NP [,] and/or NP;
6. NP, especially NP, NP [,] and/or NP;

[1] http://igm.univ-mlv.fr/~unitex/

Name	# Documents	# Tokens	# Lemmas	Size
WaCypedia	2.694.815	$2.026 \cdot 10^9$	3.368.147	5.88 Gb
ukWaC	2.694.643	$0.889 \cdot 10^9$	5.469.313	11.76 Gb
WaCypedia + ukWaC	5.387.431	$2.915 \cdot 10^9$	7.585.989	17.64 Gb

Table 1: Corpora used by the *PatternSim* measure.

```
7. NP: NP, [NP,] and/or NP;
8. NP is DET ADJ.Superl NP;
9. NP, e. g., NP, NP[,] and/or NP;
10. NP, for example, NP, NP[,] and/or NP;
11. NP, i. e.[,] NP;
12. NP (or NP);
13. NP means the same as NP;
14. NP, in other words[,] NP;
15. NP, also known as NP;
16. NP, also called NP;
17. NP alias NP;
18. NP aka NP.
```

Patterns are based on linguistic knowledge and thus provide a more precise representation than co-occurences or bag-of-word models. UNITEX makes it possible to build negative and positive contexts, to exclude meaningless adjectives, and so on. Above we presented the key features of the patterns. However, they are more complex as they take into account variation of natural language expressions. Thus, FST-based patterns can achieve higher recall than the string-based patterns such as those used by Bollegala et al. (2007).

3 Semantic Similarity Measures

The outline of the similarity measure *PatternSim* is provided in Algorithm 1. The method takes as input a set of terms of interest C. Semantic similarities between these terms are returned in a $C \times C$ sparse similarity matrix $\mathbf{S}$. An element of this matrix s_{ij} is a real number within the interval $[0; 1]$ which represents the strength of semantic similarity. The algorithm also takes as input a text corpus D.

As a first step, lexico-syntactic patterns are applied to the input corpus D (line 1). In our experiments we used three corpora: WACYPEDIA, UKWAC and the combination of both (see Table 1). Applying a cascade of FSTs to a corpus is a memory and CPU consuming operation. To make processing of these huge corpora feasible, we splited the entire corpus into blocks of 250 Mb. Processing such a block took around one hour on an Intel i5 M520@2.40GHz with 4 Gb of RAM. This is the most computationally heavy operation of Algorithm 1. The method retrieves all the concordances matching the 18 patterns. Each concordance is marked up in a specific way:

- `such {non-alcoholic [sodas]} as {[root beer]} and {[cream soda]}[PATTERN=1]`
- `{traditional[food]}, such as {[sandwich]},{[burger]}, and {[fry]}[PATTERN=2]`

Figure brackets mark the noun phrases, which are in the semantic relation; nouns and compound nouns stand between the square brackets. We extracted 1.196.468 concordances K of this type from WACYPEDIA corpus and 2.227.025 concordances from UKWAC – 3.423.493 in total.

For the next step (line 2), the nouns in the square brackets are lemmatized with the DELA dictionary[2], which consists of around 300.000 simple and 130.000 compound words. The concordances which contain at least two terms from the input vocabulary C are selected (line 3).

Subsequently, the similarity matrix $\mathbf{S}$ is filled with frequencies of pairwise extractions (line 4). At this stage, a semantic similarity score s_{ij} is equal to the number of co-occurences of terms in the square brackets within the same concordance e_{ij}. Finally, the word pairs are re-ranked with one of the methods described below (line 5):

Algorithm 1: Similarity measure *PatternSim*.

Input: Terms C, Corpus D
Output: Similarity matrix, $\mathbf{S}$ $[C \times C]$

1 $K \leftarrow extract\_concord(D)$;
2 $K_{lem} \leftarrow lemmatize\_concord(K)$;
3 $K_C \leftarrow filter\_concord(K_{lem}, C)$;
4 $\mathbf{S} \leftarrow get\_extraction\_freq(C, K)$;
5 $\mathbf{S} \leftarrow rerank(\mathbf{S}, C, D)$;
6 $\mathbf{S} \leftarrow normalize(\mathbf{S})$;
7 **return** $\mathbf{S}$;

Efreq (no re-ranking). Semantic similarity s_{ij} between c_i and c_j is equal to the frequency of extractions e_{ij} between the terms $c_i, c_j \in C$ in a set of concordances K.

Efreq-Rfreq. This formula penalizes terms that are strongly related to many words. In this case, semantic similarity of terms equals: $s_{ij} = \frac{2 \cdot \alpha \cdot e_{ij}}{e_{i*} + e_{*j}}$, where $e_{i*} = \sum_{j=1}^{|C|} e_{ij}$ is a number of

[2]Available at `http://infolingu.univ-mlv.fr/`

concordances containing word c_i and α is an expected number of semantically related words per term ($\alpha = 20$). Similarly, $e_{*j} = \sum_{i=1}^{|C|} e_{ij}$.

Efreq-Rnum. This formula also reduces the weight of terms which have many relations to other words. Here we rely on the number of extractions b_{i*} with a frequency superior to β: $b_{i*} = \sum_{j:e_{ij} \geq \beta} 1$. Semantic ranking is calculated in this case as follows: $s_{ij} = \frac{2 \cdot \mu_b \cdot e_{ij}}{b_{i*} + b_{*j}}$, where $\mu_b = \frac{1}{|C|} \sum_{i=1}^{|C|} b_{i*}$ – is an average number of related words per term and $b_{*j} = \sum_{i:e_{ij} \geq \beta} 1$. We experiment with values of $\beta \in \{1, 2, 5, 10\}$.

Efreq-Cfreq. This formula penalizes relations to general words, such as "item". According to this formula, similarity equals: $s_{ij} = \frac{P(c_i, c_j)}{P(c_i)P(c_j)}$, where $P(c_i, c_j) = \frac{e_{ij}}{\sum_{ij} e_{ij}}$ is the extraction probability of the pair $\langle c_i, c_j \rangle$, $P(c_i) = \frac{f_i}{\sum_i f_i}$ is the probability of the word c_i, and f_i is the frequency of c_i in the corpus. We use the original corpus D and the corpus of concordances K to derive f_i.

Efreq-Rnum-Cfreq. This formula combines the two previous ones: $s_{ij} = \frac{2 \cdot \mu_b}{b_{i*} + b_{*j}} \cdot \frac{P(c_i, c_j)}{P(c_i)P(c_j)}$.

Efreq-Rnum-Cfreq-Pnum. This formula integrates information to the previous one about the number of patterns $p_{ij} = \overline{1, 18}$ extracted given pair of terms $\langle c_i, c_j \rangle$. The patterns, especially (5) and (7), are prone to errors. The pairs extracted independently by several patterns are more robust than those extracted only by a single pattern. The similarity of terms equals in this case: $s_{ij} = \sqrt{p_{ij}} \cdot \frac{2 \cdot \mu_b}{b_{i*} + b_{*j}} \cdot \frac{P(c_i, c_j)}{P(c_i)P(c_j)}$.

Once the reranking is done, the similarity scores are mapped to the interval $[0; 1]$ as follows (line 6): $\acute{\mathbf{S}} = \frac{\mathbf{S} - min(\mathbf{S})}{max(\mathbf{S})}$. The method described above is implemented in an Open Source system *PatternSim* [3] (LGPLv3).

4 Evaluation and Results

We evaluated the similarity measures proposed above on three tasks – correlations with human judgements about semantic similarity, ranking of word pairs and extraction of semantic relations. [4]

[3] https://github.com/cental/PatternSim

[4] Evaluation scripts and the results: http://cental.fltr.ucl.ac.be/team/panchenko/sim-eval

4.1 Correlation with Human Judgements

We use three standard human judgement datasets – MC (Miller and Charles, 1991), RG (Rubenstein and Goodenough, 1965) and WordSim353 (Finkelstein et al., 2001), composed of 30, 65, and 353 pairs of terms respectively. The quality of a measure is assessed with Spearman's correlation between vectors of scores.

The first three columns of Table 2 present the correlations. The first part of the table reports on scores of 12 baseline similarity measures: three WordNet-based (*WuPalmer, Lecock-Chodorow*, and *Resnik*), three corpus-based (*ContextWindow, SyntacticContext,* and *LSA*), three definition-based (*WiktionaryOverlap, GlossVectors,* and *ExtendedLesk*), and three *WikiRelate* measures. The second part of the table presents various modifications of our measure based on lexico-syntactic patterns. The first two are based on WACKY and UKWAC corpora, respectively. All the remaining *PatternSim* measures are based on both corpora (WACKY+UKWAC) as, according to our experiments, they provide better results. Correlations of measures based on patterns are comparable to those of the baselines. In particular, *PatternSim* performs similarly to the measures based on WordNet and dictionary glosses, but requires no hand-crafted resources. Furthermore, the proposed measures outperform most of the baselines on the WordSim353 dataset achieving a correlation of 0.520.

4.2 Semantic Relation Ranking

In this task, a similarity measure is used to rank pairs of terms. Each "target" term has roughly the same number of meaningful and random "relatums". A measure should rank semantically similar pairs higher than the random ones. We use two datasets: BLESS (Baroni and Lenci, 2011) and SN (Panchenko and Morozova, 2012). BLESS relates 200 target nouns to 8625 relatums with 26.554 semantic relations (14.440 are meaningful and 12.154 are random) of the following types: hypernymy, co-hyponymy, meronymy, attribute, event, or random. SN relates 462 target nouns to 5.910 relatum with 14.682 semantic relations (7.341 are meaningful and 7.341 are random) of the following types: synonymy, hypernymy, co-hyponymy, and random. Let R be a set of cor-

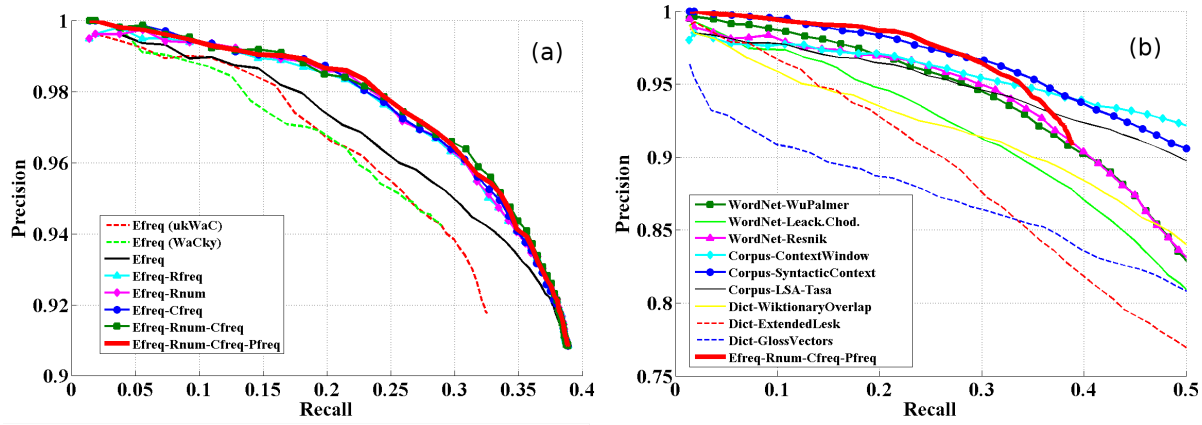

Figure 1: Precision-Recall graphs calculated on the BLESS (hypo,cohypo,mero,attri,event) dataset: (a) variations of the *PatternSim* measure; (b) the best *PatternSim* measure as compared to the baseline similarity measures.

rect relations and $\hat{R}_k$ be a set of semantic relations among the top $k\%$ nearest neigbors of target terms. Then precision and recall at k are defined as follows: $P(k) = \frac{|R \cap \hat{R}_k|}{|\hat{R}_k|}, R(k) = \frac{|R \cap \hat{R}_k|}{|R|}$. The quality of a measure is assessed with $P(10)$, $P(20)$, $P(50)$, and $R(50)$.

Table 2 and Figure 1 present the performance of baseline and pattern-based measures on these datasets. Precision of the similarity scores learned from the WACKY corpus is higher than that obtained from the UKWAC, but recall of UKWAC is better since this corpus is bigger (see Figure 1 (a)). Thus, in accordance with the previous evaluation, the biggest corpus WACKY+UKWAC provides better results than the WACKY or the UKWAC alone. Ranking relations with extraction frequencies (*Efreq*) provides results that are significantly worse than any re-ranking strategies. On the other hand, the difference between various re-ranking formulas is small with a slight advantage for *Efreq-Rnum-Cfreq-Pnum*.

The performance of the *Efreq-Rnum-Cfreq-Pnum* measure is comparable to the baselines (see Figure 1 (b)). Furthermore, in terms of precision, it outperforms the 9 baselines, including syntactic distributional analysis (*Corpus-SyntacticContext*). However, its recall is seriously lower than the baselines because of the sparsity of the pattern-based approach. The similarity of terms can only be calculated if they co-occur in the corpus within an extraction pattern. Contrastingly, *PatternSim* achieves both high recall and precision on BLESS dataset containing only hyponyms and co-hyponyms (see Table 2).

4.3 Semantic Relation Extraction

We evaluated relations extracted with the *Efreq* and the *Efreq-Rnum-Cfreq-Pnum* measures for 49 words (vocabulary of the RG dataset). Three annotators indicated whether the terms were semantically related or not. We calculated for each of 49 words extraction precision at $k = \{1, 5, 10, 20, 50\}$. Figure 2 shows the results of this evaluation. For the *Efreq* measure, average precision indicated by white squares varies between 0.792 (the top relation) and 0.594 (the 20 top relations), whereas it goes from 0.736 (the top relation) to 0.599 (the 20 top relations) for the *Efreq-Rnum-Cfreq-Pnum* measure. The inter-raters agreement (Fleiss's kappa) is substantial (0.61-0.80) or moderate (0.41-0.60).

5 Conclusion

In this work, we presented a similarity measure based on manually-crafted lexico-syntactic patterns. The measure was evaluated on five ground truth datasets (MC, RG, WordSim353, BLESS, SN) and on the task of semantic relation extraction. Our results have shown that the measure provides results comparable to the baseline WordNet-, dictionary-, and corpus-based measures and does not require semantic resources.

In future work, we are going to use a logistic regression to choose parameter values (α and β) and to combine different factors ($e_{ij}, e_{i*}, P(c_i), P(c_i, c_j), p_{ij}$, etc.) in one model.

Similarity Measure	MC	RG	WS	BLESS (hypo,cohypo,mero,attri,event)				SN (syn, hypo, cohypo)				BLESS (hypo, cohypo)			
	ρ	*ρ*	*ρ*	*P(10)*	*P (20)*	*P(50)*	*R(50)*	*P(10)*	*P(20)*	*P(50)*	*R(50)*	*P(10)*	*P(20)*	*P(50)*	*R(50)*
Random	*0.056*	*-0.047*	*-0.122*	0.546	0.542	0.544	0.522	0.504	0.502	0.499	0.498	0.271	0.279	0.286	0.502
WordNet-WuPalmer	0.742	0.775	0.331	0.974	0.929	0.702	0.674	0.982	0.959	0.766	**0.763**	0.977	0.932	0.547	**0.968**
WordNet-Leack.Chod.	0.724	**0.789**	0.295	0.953	0.901	0.702	0.648	**0.984**	0.953	0.757	0.755	0.951	0.897	0.542	0.957
WordNet-Resnik	0.784	0.757	0.331	0.970	0.933	0.700	0.647	0.948	0.908	0.724	0.722	**0.968**	0.938	0.542	0.956
Corpus-ContextWindow	0.693	0.782	0.466	0.971	0.947	0.836	**0.772**	0.974	0.932	0.742	0.740	0.908	0.828	0.502	0.886
Corpus-SynContext	**0.790**	0.786	0.491	0.985	0.953	0.811	0.749	0.978	0.945	0.751	0.743	0.979	0.921	0.536	0.947
Corpus-LSA-Tasa	0.694	0.605	0.566	0.968	0.937	0.802	0.740	0.903	0.846	0.641	0.609	0.877	0.775	0.467	0.824
Dict-WiktionaryOverlap	0.759	0.754	0.521	0.943	0.905	0.750	0.679	0.922	0.887	0.725	0.656	0.837	0.769	0.518	0.739
Dict-GlossVectors	0.653	0.738	0.322	0.894	0.860	0.742	0.686	0.932	0.899	0.722	0.709	0.777	0.702	0.449	0.793
Dict-ExtenedLesk	0.792	0.718	0.409	0.937	0.866	0.711	0.657	0.952	0.873	0.655	0.654	0.873	0.751	0.464	0.820
WikiRelate-Gloss	0.460	0.460	0.200	–	–	–	–	–	–	–	–	–	–	–	–
WikiRelate-Leack.Chod.	0.410	0.500	0.480	–	–	–	–	–	–	–	–	–	–	–	–
WikiRelate-SVM	–	–	**0.590**	–	–	–	–	–	–	–	–	–	–	–	–
Efreq (WaCky)	0.522	0.574	0.405	0.971	0.950	0.942	0.289	0.930	0.912	0.897	0.306	0.976	0.937	0.923	0.626
Efreq (ukWaC)	0.384	0.562	0.411	0.974	0.944	0.918	0.325	0.922	0.905	0.869	0.329	0.971	0.926	0.884	0.653
Efreq	0.486	0.632	0.429	0.980	0.945	0.909	**0.389**	0.938	0.915	0.866	**0.400**	0.976	0.929	0.865	**0.739**
Efreq-Rfreq	**0.666**	**0.739**	0.508	0.987	0.955	0.909	**0.389**	0.951	0.922	0.867	**0.400**	0.983	0.940	0.865	**0.739**
Efreq-Rnum	0.647	0.720	0.499	0.989	0.955	0.909	**0.389**	0.951	0.922	0.867	**0.400**	0.983	0.940	0.865	**0.739**
Efreq-Cfreq	0.600	0.709	0.493	0.989	0.956	0.909	**0.389**	0.949	0.920	0.867	**0.400**	0.986	0.948	0.865	**0.739**
Efreq-Cfreq (concord.)	**0.666**	**0.739**	0.508	0.986	0.954	0.909	**0.389**	0.952	0.921	0.867	**0.400**	0.984	0.944	0.865	**0.739**
Efreq-Rnum-Cfreq	0.647	0.737	0.513	0.988	0.959	0.909	**0.389**	**0.953**	0.924	0.867	**0.400**	**0.987**	0.947	0.865	**0.739**
Efreq-Rnum-Cfreq-Pnum	0.647	0.737	**0.520**	**0.989**	0.957	0.909	**0.389**	0.952	0.924	0.867	**0.400**	0.985	0.947	0.865	**0.739**

Table 2: Performance of the baseline similarity measures as compared to various modifications of the *PatternSim* measure on human judgements datasets (MC, RG, WS) and semantic relation datasets (BLESS and SN).

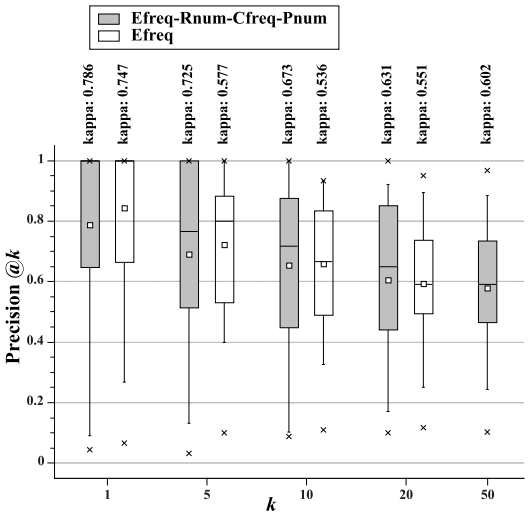

Figure 2: Semantic relation extraction: precision at k.

References

S. Banerjee and T. Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI*, volume 18, pages 805–810.

M. Baroni and A. Lenci. 2011. How we blessed distributional semantic evaluation. In *GEMS (EMNLP), 2011*, pages 1–11.

D. Bollegala, Y. Matsuo, and M. Ishizuka. 2007. Measuring semantic similarity between words using web search engines. In *WWW*, volume 766.

L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin. 2001. Placing search in context: The concept revisited. In *WWW 2001*, pages 406–414.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *ACL*, pages 539–545.

T. K. Landauer, P. W. Foltz, and D. Laham. 1998. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284.

C. Leacock and M. Chodorow. 1998. Combining Local Context and WordNet Similarity for Word Sense Identification. *WordNet*, pages 265–283.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *ACL*, pages 768–774.

G. A. Miller. 1995. Wordnet: a lexical database for english. *Communications of ACM*, 38(11):39–41.

A. Panchenko and O. Morozova. 2012. A study of hybrid similarity measures for semantic relation extraction. *Hybrid Approaches to the Processing of Textual Data (EACL)*, pages 10–18.

S. Patwardhan and T. Pedersen. 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. *Making Sense of Sense: Bringing Psycholinguistics and Computational Linguistics Together*, pages 1–12.

P. Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *IJCAI*, volume 1, pages 448–453.

M. Strube and S. P. Ponzetto. 2006. Wikirelate! computing semantic relatedness using wikipedia. In *AAAI*, volume 21, pages 14–19.

T. Van de Cruys. 2010. *Mining for Meaning: The Extraction of Lexico-Semantic Knowledge from Text*. Ph.D. thesis, University of Groningen.

Z. Wu and M. Palmer. 1994. Verbs semantics and lexical selection. In *ACL'1994*, pages 133–138.

T. Zesch, C. Müller, and I. Gurevych. 2008. Extracting lexical semantic knowledge from wikipedia and wiktionary. In *LREC'08*, pages 1646–1652.