

A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches

Eneko Agirre[†] Enrique Alfonseca[‡] Keith Hall[‡] Jana Kravalova^{‡§} Marius Paşca[‡] Aitor Soroa[†]

[†] IXA NLP Group, University of the Basque Country

[‡] Google Inc.

[§] Institute of Formal and Applied Linguistics, Charles University in Prague

{e.agirre, a.soroa}@ehu.es {ealfonseca, kbhall, mars}@google.com

kravalova@ufal.mff.cuni.cz

Abstract

This paper presents and compares WordNet-based and distributional similarity approaches. The strengths and weaknesses of each approach regarding similarity and relatedness tasks are discussed, and a combination is presented. Each of our methods independently provide the best results in their class on the RG and WordSim353 datasets, and a supervised combination of them yields the best published results on all datasets. Finally, we pioneer cross-lingual similarity, showing that our methods are easily adapted for a cross-lingual task with minor losses.

1 Introduction

Measuring semantic similarity and relatedness between terms is an important problem in lexical semantics. It has applications in many natural language processing tasks, such as Textual Entailment, Word Sense Disambiguation or Information Extraction, and other related areas like Information Retrieval. The techniques used to solve this problem can be roughly classified into two main categories: those relying on pre-existing knowledge resources (thesauri, semantic networks, taxonomies or encyclopedias) (Alvarez and Lim, 2007; Yang and Powers, 2005; Hughes and Ramage, 2007) and those inducing distributional properties of words from corpora (Sahami and Heilman, 2006; Chen et al., 2006; Bollegala et al., 2007).

In this paper, we explore both families. For the first one we apply graph based algorithms to WordNet, and for the second we induce distributional similarities collected from a 1.6 Terabyte Web corpus. Previous work suggests that distributional similarities suffer from certain limitations, which make

them less useful than knowledge resources for semantic similarity. For example, Lin (1998b) finds similar phrases like *captive-westerner* which made sense only in the context of the corpus used, and Budanitsky and Hirst (2006) highlight other problems that stem from the imbalance and sparseness of the corpora. Comparatively, the experiments in this paper demonstrate that distributional similarities can perform as well as the knowledge-based approaches, and a combination of the two can exceed the performance of results previously reported on the same datasets. An application to cross-lingual (CL) similarity identification is also described, with applications such as CL Information Retrieval or CL sponsored search. A discussion on the differences between learning similarity and relatedness scores is provided.

The paper is structured as follows. We first present the WordNet-based method, followed by the distributional methods. Section 4 is devoted to the evaluation and results on the monolingual and cross-lingual tasks. Section 5 presents some analysis, including learning curves for distributional methods, the use of distributional similarity to improve WordNet similarity, the contrast between similarity and relatedness, and the combination of methods. Section 6 presents related work, and finally, Section 7 draws the conclusions and mentions future work.

2 WordNet-based method

WordNet (Fellbaum, 1998) is a lexical database of English, which groups nouns, verbs, adjectives and adverbs into sets of synonyms (synsets), each expressing a distinct concept. Synsets are interlinked with conceptual-semantic and lexical relations, including hypernymy, meronymy, causality, etc.

Given a pair of words and a graph-based representation of WordNet, our method has basically two

steps: We first compute the personalized PageRank over WordNet separately for each of the words, producing a probability distribution over WordNet synsets. We then compare how similar these two discrete probability distributions are by encoding them as vectors and computing the cosine between the vectors.

We represent WordNet as a graph $G = (V, E)$ as follows: graph nodes represent WordNet concepts (synsets) and dictionary words; relations among synsets are represented by undirected edges; and dictionary words are linked to the synsets associated to them by directed edges.

For each word in the pair we first compute a personalized PageRank vector of graph G (Haveliwala, 2002). Basically, personalized PageRank is computed by modifying the random jump distribution vector in the traditional PageRank equation. In our case, we concentrate all probability mass in the target word.

Regarding PageRank implementation details, we chose a damping value of 0.85 and finish the calculation after 30 iterations. These are default values, and we did not optimize them. Our similarity method is similar, but simpler, to that used by (Hughes and Ramage, 2007), which report very good results on similarity datasets. More details of our algorithm can be found in (Agirre and Soroa, 2009). The algorithm and needed resources are publicly available¹.

2.1 WordNet relations and versions

The WordNet versions that we use in this work are the Multilingual Central Repository or MCR (Atserias et al., 2004) (which includes English WordNet version 1.6 and wordnets for several other languages like Spanish, Italian, Catalan and Basque), and WordNet version 3.0². We used all the relations in MCR (except cooccurrence relations and selectional preference relations) and in WordNet 3.0. Given the recent availability of the disambiguated gloss relations for WordNet 3.0³, we also used a version which incorporates these relations. We will refer to the three versions as MCR16, WN30 and WN30g, respectively. Our choice was mainly motivated by the fact that MCR contains tightly aligned

wordnets of several languages (see below).

2.2 Cross-linguality

MCR follows the EuroWordNet design (Vossen, 1998), which specifies an InterLingual Index (ILI) that links the concepts across wordnets of different languages. The wordnets for other languages in MCR use the English WordNet synset numbers as ILIs. This design allows a decoupling of the relations between concepts (which can be taken to be language independent) and the links from each content word to its corresponding concepts (which is language dependent).

As our WordNet-based method uses the graph of the concepts and relations, we can easily compute the similarity between words from different languages. For example, consider a English-Spanish pair like *car* – *coche*. Given that the Spanish WordNet is included in MCR we can use MCR as the common knowledge-base for the relations. We can then compute the personalized PageRank for each of *car* and *coche* on the same underlying graph, and then compare the similarity between both probability distributions.

As an alternative, we also tried to use publicly available mappings for wordnets (Daude et al., 2000)⁴ in order to create a 3.0 version of the Spanish WordNet. The mapping was used to link Spanish variants to 3.0 synsets. We used the English WordNet 3.0, including glosses, to construct the graph. The two Spanish WordNet versions are referred to as MCR16 and WN30g.

3 Context-based methods

In this section, we describe the distributional methods used for calculating similarities between words, and profiting from the use of a large Web-based corpus.

This work is motivated by previous studies that make use of search engines in order to collect co-occurrence statistics between words. Turney (2001) uses the number of hits returned by a Web search engine to calculate the Pointwise Mutual Information (PMI) between terms, as an indicator of synonymy. Bollegala et al. (2007) calculate a number of popular relatedness metrics based on page counts,

¹<http://http://ixa2.si.ehu.es/ukb/>

²Available from <http://http://wordnet.princeton.edu/>

³<http://wordnet.princeton.edu/glosstag>

⁴<http://www.lsi.upc.es/~nlp/tools/download-map.php>

like PMI, the Jaccard coefficient, the Simpson coefficient and the Dice coefficient, which are combined with lexico-syntactic patterns as model features. The model parameters are trained using Support Vector Machines (SVM) in order to later rank pairs of words. A different approach is the one taken by Sahami and Heilman (2006), who collect snippets from the results of a search engine and represent each snippet as a vector, weighted with the tf-idf score. The semantic similarity between two queries is calculated as the inner product between the centroids of the respective sets of vectors.

To calculate the similarity of two words w_1 and w_2 , Ruiz-Casado et al. (2005) collect snippets containing w_1 from a Web search engine, extract a context around it, replace it with w_2 and check for the existence of that modified context in the Web.

Using a search engine to calculate similarities between words has the drawback that the data used will always be truncated. So, for example, the numbers of hits returned by search engines nowadays are always approximate and rounded up. The systems that rely on collecting snippets are also limited by the maximum number of documents returned per query, typically around a thousand. We hypothesize that by crawling a large corpus from the Web and doing standard corpus analysis to collect precise statistics for the terms we should improve over other unsupervised systems that are based on search engine results, and should yield results that are competitive even when compared to knowledge-based approaches.

In order to calculate the semantic similarity between the words in a set, we have used a vector space model, with the following three variations:

In the **bag-of-words approach**, for each word w in the dataset we collect every term t that appears in a window centered in w , and add them to the vector together with its frequency.

In the **context window approach**, for each word w in the dataset we collect every window W centered in w (removing the central word), and add it to the vector together with its frequency (the total number of times we saw window W around w in the whole corpus). In this case, all punctuation symbols are replaced with a special token, to unify patterns like *, the <term> said to* and *' the <term> said to*. Throughout the paper, when we mention a context

window of size N it means N words at each side of the phrase of interest.

In the **syntactic dependency approach**, we parse the entire corpus using an implementation of an Inductive Dependency parser as described in Nivre (2006). For each word w we collect a template of the syntactic context. We consider sequences of governing words (e.g. the parent, grand-parent, etc.) as well as collections of descendants (e.g., immediate children, grandchildren, etc.). This information is then encoded as a contextual template. For example, the context template *cooks <term> delicious* could be contexts for nouns such as *food, meals, pasta*, etc. This captures both syntactic preferences as well as selectional preferences. Contrary to Pado and Lapata (2007), we do not use the labels of the syntactic dependencies.

Once the vectors have been obtained, the frequency for each dimension in every vector is weighted using the other vectors as contrast set, with the χ^2 test, and finally the cosine similarity between vectors is used to calculate the similarity between each pair of terms.

Except for the syntactic dependency approach, where closed-class words are needed by the parser, in the other cases we have removed stopwords (pronouns, prepositions, determiners and modal and auxiliary verbs).

3.1 Corpus used

We have used a corpus of four billion documents, crawled from the Web in August 2008. An HTML parser is used to extract text, the language of each document is identified, and non-English documents are discarded. The final corpus remaining at the end of this process contains roughly 1.6 Terawords. All calculations are done in parallel sharding by dimension, and it is possible to calculate all pairwise similarities of the words in the test sets very quickly on this corpus using the MapReduce infrastructure. A complete run takes around 15 minutes on 2,000 cores.

3.2 Cross-linguality

In order to calculate similarities in a cross-lingual setting, where some of the words are in a language l other than English, the following algorithm is used:

Method	Window size	RG dataset	WordSim353 dataset
MCR16		0.83 [0.73, 0.89]	0.53 (0.56) [0.45, 0.60]
WN30		0.79 [0.67, 0.86]	0.56 (0.58) [0.48, 0.63]
WN30g		0.83 [0.73, 0.89]	0.66 (0.69) [0.59, 0.71]
CW	1	0.83 [0.73, 0.89]	0.63 [0.57, 0.69]
	2	0.83 [0.74, 0.90]	0.60 [0.53, 0.66]
	3	0.85 [0.76, 0.91]	0.59 [0.52, 0.65]
	4	0.89 [0.82, 0.93]	0.60 [0.53, 0.66]
	5	0.80 [0.70, 0.88]	0.58 [0.51, 0.65]
	6	0.75 [0.62, 0.84]	0.58 [0.50, 0.64]
	7	0.72 [0.58, 0.82]	0.57 [0.49, 0.63]
BoW	1	0.81 [0.70, 0.88]	0.64 [0.57, 0.70]
	2	0.80 [0.69, 0.87]	0.64 [0.58, 0.70]
	3	0.79 [0.67, 0.86]	0.64 [0.58, 0.70]
	4	0.78 [0.66, 0.86]	0.65 [0.58, 0.70]
	5	0.77 [0.64, 0.85]	0.64 [0.58, 0.70]
	6	0.76 [0.63, 0.85]	0.65 [0.58, 0.70]
	7	0.75 [0.62, 0.84]	0.64 [0.58, 0.70]
Syn	G1,D0	0.81 [0.70, 0.88]	0.62 [0.55, 0.68]
	G2,D0	0.82 [0.72, 0.89]	0.55 [0.48, 0.62]
	G3,D0	0.81 [0.71, 0.88]	0.62 [0.56, 0.68]
	G1,D1	0.82 [0.72, 0.89]	0.62 [0.55, 0.68]
	G2,D1	0.82 [0.73, 0.89]	0.62 [0.55, 0.68]
	G3,D1	0.82 [0.72, 0.88]	0.62 [0.55, 0.68]
CW+	4; G1,D0	0.88 [0.81, 0.93]	0.66 [0.59, 0.71]
Syn	4; G2,D0	0.87 [0.80, 0.92]	0.64 [0.57, 0.70]
	4; G3,D0	0.86 [0.77, 0.91]	0.63 [0.56, 0.69]
	4; G1,D1	0.83 [0.73, 0.89]	0.48 [0.40, 0.56]
	4; G2,D1	0.83 [0.73, 0.89]	0.49 [0.40, 0.56]
	4; G3,D1	0.82 [0.72, 0.89]	0.48 [0.40, 0.56]

Table 1: Spearman correlation results for the various WordNet-based models and distributional models. CW=Context Windows, BoW=bag of words, Syn=syntactic vectors. For Syn, the *window size* is actually the tree-depth for the governors and descendants. For examples, G1 indicates that the contexts include the parents and D2 indicates that both the children and grandchildren make up the contexts. The final grouping includes both contextual windows (at width 4) and syntactic contexts in the template vectors. Max scores are bolded.

1. Replace each non-English word in the dataset with its 5-best translations into English using state-of-the-art machine translation technology.
2. The vector corresponding to each Spanish word is calculated by collecting features from all the contexts of any of its translations.
3. Once the vectors are generated, the similarities are calculated in the same way as before.

4 Experimental results

4.1 Gold-standard datasets

We have used two standard datasets. The first one, RG, consists of 65 pairs of words collected by Rubenstein and Goodenough (1965), who had them judged by 51 human subjects in a scale from 0.0 to 4.0 according to their similarity, but ignoring any other possible semantic relationships that might appear between the terms. The second dataset, WordSim353⁵ (Finkelstein et al., 2002) contains 353 word pairs, each associated with an average of 13 to 16 human judgements. In this case, both similarity and re-

⁵ Available at <http://www.cs.technion.ac.il/~gabr/resources/data/wordsim353/wordsim353.html>

Context	RG terms and frequencies
Il never forget the * on his face when	grin,2,smile,10
he had a giant * on his face and	grin,3,smile,2
room with a huge * on her face and	grin,2,smile,6
the state of every * will be updated every	automobile,2,car,3
repair or replace the * if it is stolen	shore,14,coast,2
located on the north * of the Bay of	shore,3,coast,2
areas on the eastern * of the Adriatic Sea	slave,3,boy,5,shore,3,string,2
Thesaurus of Current English * The Oxford Pocket Thesaurus	wizard,4,glass,4,crane,5,smile,5
	implement,5,oracle,2,lad,2
	food,3,car,2,madhouse,3,jewel,3
	asylum,4,tool,8,journey,6,etc.
be understood that the * 10 may be designed	crane,3,tool,3
a fight between a * and a snake and	bird,3,crane,5

Table 2: Sample of context windows for the terms in the RG dataset.

latedness are annotated without any distinction. Several studies indicate that the human scores consistently have very high correlations with each other (Miller and Charles, 1991; Resnik, 1995), thus validating the use of these datasets for evaluating semantic similarity.

For the cross-lingual evaluation, the two datasets were modified by translating the second word in each pair into Spanish. Two humans translated simultaneously both datasets, with an inter-tagger agreement of 72% for RG and 84% for WordSim353.

4.2 Results

Table 1 shows the Spearman correlation obtained on the RG and WordSim353 datasets, including the interval at 0.95 of confidence⁶.

Overall the distributional context-window approach performs best in the RG, reaching 0.89 correlation, and both WN30g and the combination of context windows and syntactic context perform best on WordSim353. Note that the confidence intervals are quite large in both RG and WordSim353, and few of the pairwise differences are statistically significant.

Regarding WordNet-based approaches, the use of the glosses and WordNet 3.0 (WN30g) yields the best results in both datasets. While MCR16 is close to WN30g for the RG dataset, it lags well behind on WordSim353. This discrepancy is further analyzed in Section 5.3. Note that the performance of WordNet in the WordSim353 dataset suffers from unknown words. In fact, there are nine pairs which returned null similarity for this reason. The num-

⁶To calculate the Spearman correlations values are transformed into ranks, and we calculate the Pearson correlation on them. The confidence intervals refer to the Pearson correlations of the rank vectors.

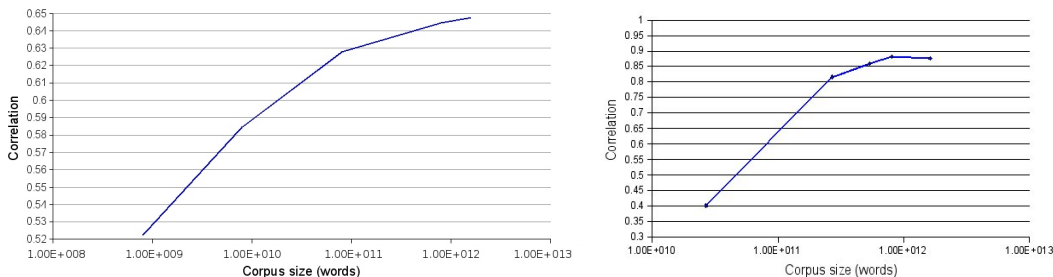


Figure 1: Effect of the size of the training corpus, for the best distributional similarity model in each dataset. Left: WordSim353 with bag-of-words, Right: RG with context windows.

Dataset	Method	overall	Δ	interval
RG	MCR16	0.78	-0.05	[0.66, 0.86]
	WN30g	0.74	-0.09	[0.61, 0.84]
	Bag of words	0.68	-0.23	[0.53, 0.79]
	Context windows	0.83	-0.05	[0.73, 0.89]
WS353	MCR16	0.42 (0.53)	-0.11 (-0.03)	[0.34, 0.51]
	WN30g	0.58 (0.67)	-0.07 (-0.02)	[0.51, 0.64]
	Bag of words	0.53	-0.12	[0.45, 0.61]
	Context windows	0.52	-0.11	[0.44, 0.59]

Table 3: Results obtained by the different methods on the Spanish/English cross-lingual datasets. The Δ column shows the performance difference with respect to the results on the original dataset.

ber in parenthesis in Table 1 for WordSim353 shows the results for the 344 remaining pairs. Section 5.2 shows a proposal to overcome this limitation.

The bag-of-words approach tends to group together terms that can have a similar distribution of contextual terms. Therefore, terms that are topically related can appear in the same textual passages and will get high values using this model. We see this as an explanation why this model performed better than the context window approach for WordSim353, where annotators were instructed to provide high ratings to related terms. On the contrary, the context window approach tends to group together words that are exchangeable in exactly the same context, preserving order. Table 2 illustrates a few examples of context collected. Therefore, true synonyms and hyponyms/hyperonyms will receive high similarities, whereas terms related topically or based on any other semantic relation (e.g. *movie* and *star*) will have lower scores. This explains why this method performed better for the RG dataset. Section 5.3 confirms these observations.

4.3 Cross-lingual similarity

Table 3 shows the results for the English-Spanish cross-lingual datasets. For RG, MCR16 and the

context windows methods drop only 5 percentage points, showing that cross-lingual similarity is feasible, and that both cross-lingual strategies are robust.

The results for WordSim353 show that WN30g is the best for this dataset, with the rest of the methods falling over 10 percentage points relative to the monolingual experiment. A closer look at the WordNet results showed that most of the drop in performance was caused by out-of-vocabulary words, due to the smaller vocabulary of the Spanish WordNet. Though not totally comparable, if we compute the correlation over pairs covered in WordNet alone, the correlation would drop only 2 percentage points. In the case of the distributional approaches, the fall in performance was caused by the translations, as only 61% of the words were translated into the original word in the English datasets.

5 Detailed analysis and system combination

In this section we present some analysis, including learning curves for distributional methods, the use of distributional similarity to improve WordNet similarity, the contrast between similarity and relatedness, and the combination of methods.

5.1 Learning curves for distributional methods

Figure 1 shows that the correlation improves with the size of the corpus, as expected. For the results using the WordSim353 corpus, we show the results of the bag-of-words approach with context size 10. Results improve from 0.5 Spearman correlation up to 0.65 when increasing the corpus size three orders of magnitude, although the effect decays at the end, which indicates that we might not get fur-

Method	Without similar words	With similar words
WN30	0.56 (0.58) [0.48, 0.63]	0.58 [0.51, 0.65]
WN30g	0.66 (0.69) [0.59, 0.71]	0.68 [0.62, 0.73]

Table 4: Results obtained replacing unknown words with their most similar three words (WordSim353 dataset).

Method	overall	Similarity	Relatedness
MCR16	0.53 [0.45, 0.60]	0.65 [0.56, 0.72]	0.33 [0.21, 0.43]
WN30	0.56 [0.48, 0.63]	0.73 [0.65, 0.79]	0.38 [0.27, 0.48]
WN30g	0.66 [0.59, 0.71]	0.72 [0.64, 0.78]	0.56 [0.46, 0.64]
BoW	0.65 [0.59, 0.71]	0.70 [0.63, 0.77]	0.62 [0.53, 0.69]
CW	0.60 [0.53, 0.66]	0.77 [0.71, 0.82]	0.46 [0.36, 0.55]

Table 5: Results obtained on the WordSim353 dataset and on the two similarity and relatedness subsets.

ther gains going beyond the current size of the corpus. With respect to results for the RG dataset, we used a context-window approach with context radius 4. Here, results improve even more with data size, probably due to the sparse data problem collecting 8-word context windows if the corpus is not large enough. Correlation improves linearly right to the end, where results stabilize around 0.89.

5.2 Combining both approaches: dealing with unknown words in WordNet

Although the vocabulary of WordNet is very extensive, applications are bound to need the similarity between words which are not included in WordNet. This is exemplified in the WordSim353 dataset, where 9 pairs contain words which are unknown to WordNet. In order to overcome this shortcoming, we could use similar words instead, as provided by the distributional thesaurus. We used the distributional thesaurus defined in Section 3, using context windows of width 4, to provide three similar words for each of the unknown words in WordNet. Results improve for both WN30 and WN30g, as shown in Table 4, attaining our best results for WordSim353.

5.3 Similarity vs. relatedness

We mentioned above that the annotation guidelines of WordSim353 did not distinguish between similar and related pairs. As the results in Section 4 show, different techniques are more appropriate to calculate either similarity or relatedness. In order to study this effect, ideally, we would have two versions of the dataset, where annotators were given precise instructions to distinguish similarity in one case, and relatedness in the other. Given the lack of such datasets, we devised a simpler approach in

order to reuse the existing human judgements. We manually split the dataset in two parts, as follows.

First, two humans classified all pairs as being synonyms of each other, antonyms, identical, hyperonym-hyponym, hyponym-hyperonym, holonym-meronym, meronym-holonym, and none-of-the-above. The inter-tagger agreement rate was 0.80, with a Kappa score of 0.77. This annotation was used to group the pairs in three categories: similar pairs (those classified as synonyms, antonyms, identical, or hyponym-hyperonym), related pairs (those classified as meronym-holonym, and pairs classified as none-of-the-above, with a human average similarity greater than 5), and unrelated pairs (those classified as none-of-the-above that had average similarity less than or equal to 5). We then created two new gold-standard datasets: **similarity** (the union of similar and unrelated pairs), and **relatedness** (the union of related and unrelated)⁷.

Table 5 shows the results on the relatedness and similarity subsets of WordSim353 for the different methods. Regarding WordNet methods, both WN30 and WN30g perform similarly on the similarity subset, but WN30g obtains the best results by far on the relatedness data. These results are congruent with our expectations: two words are similar if their synsets are in close places in the WordNet hierarchy, and two words are related if there is a connection between them. Most of the relations in WordNet are of hierarchical nature, and although other relations exist, they are far less numerous, thus explaining the good results for both WN30 and WN30g on similarity, but the bad results of WN30 on relatedness. The disambiguated glosses help find connections among related concepts, and allow our method to better model relatedness with respect to WN30.

The low results for MCR16 also deserve some comments. Given the fact that MCR16 performed very well on the RG dataset, it comes as a surprise that it performs so poorly for the similarity subset of WordSim353. In an additional evaluation, we attested that MCR16 does indeed perform as well as MCR30g on the **similar pairs** subset. We believe that this deviation could be due to the method used to construct the similarity dataset, which includes some pairs of loosely related pairs labeled as unrelated.

⁷Available at <http://alfonseca.org/eng/research/wordsim353.html>

Methods combined in the SVM	RG dataset	WordSim353 dataset	WordSim353 similarity	WordSim353 relatedness
WN30g, bag of words	0.88 [0.82, 0.93]	0.78 [0.73, 0.81]	0.81 [0.76, 0.86]	0.72 [0.65, 0.77]
WN30g, context windows	0.90 [0.84, 0.94]	0.73 [0.68, 0.79]	0.83 [0.78, 0.87]	0.64 [0.56, 0.71]
WN30g, syntax	0.89 [0.83, 0.93]	0.75 [0.70, 0.79]	0.83 [0.78, 0.87]	0.67 [0.60, 0.74]
WN30g, bag of words, context windows, syntax	0.96 [0.93, 0.97]	0.78 [0.73, 0.82]	0.83 [0.78, 0.87]	0.71 [0.65, 0.77]

Table 6: Results using a supervised combination of several systems. Max values are bolded for each dataset.

Concerning the techniques based on distributional similarities, the method based on context windows provides the best results for similarity, and the bag-of-words representation outperforms most of the other techniques for relatedness.

5.4 Supervised combination

In order to gain an insight on which would be the upper bound that we could obtain when combining our methods, we took the output of three systems (bag of words with window size 10, context window with size 4, and the WN30g run). Each of these outputs is a ranking of word pairs, and we implemented an oracle that chooses, for each pair, the rank that is most similar to the rank of the pair in the gold-standard. The outputs of the oracle have a Spearman correlation of 0.97 for RG and 0.92 for WordSim353, which gives as an indication of the correlations that could be achieved by choosing for each pair the rank output by the best classifier for that pair.

The previous results motivated the use of a supervised approach to combine the output of the different systems. We created a training corpus containing pairs of pairs of words from the datasets, having as features the similarity and rank of each pair involved as given by the different unsupervised systems. A classifier is trained to decide whether the first pair is more similar than the second one. For example, a training instance using two unsupervised classifiers is 0.001364, 31, 0.327515, 64, 0.084805, 57, 0.109061, 59, *negative* meaning that the similarities given by the first classifier to the two pairs were 0.001364 and 0.327515 respectively, which ranked them in positions 31 and 64. The second classifier gave them similarities of 0.084805 and 0.109061 respectively, which ranked them in positions 57 and 59. The class *negative* indicates that in the gold-standard the first pair has a lower score than the second pair.

We have trained a SVM to classify pairs of pairs, and use its output to rank the entries in both datasets. It uses a polynomial kernel with degree 4. We did

Method	Source	Spearman (MC)	Pearson (MC)
(Sahami et al., 2006)	Web snippets	0.62 [0.32, 0.81]	0.58 [0.26, 0.78]
(Chen et al., 2006)	Web snippets	0.69 [0.42, 0.84]	0.69 [0.42, 0.85]
(Wu and Palmer, 1994)	WordNet	0.78 [0.59, 0.90]	0.78 [0.57, 0.89]
(Leacock et al., 1998)	WordNet	0.79 [0.59, 0.90]	0.82 [0.64, 0.91]
(Resnik, 1995)	WordNet	0.81 [0.62, 0.91]	0.80 [0.60, 0.90]
(Lin, 1998a)	WordNet	0.82 [0.65, 0.91]	0.83 [0.67, 0.92]
(Bollegala et al., 2007)	Web snippets	0.82 [0.64, 0.91]	0.83 [0.67, 0.92]
(Jiang and Conrath, 1997)	WordNet	0.83 [0.67, 0.92]	0.85 [0.69, 0.93]
(Jarmasz, 2003)	Roget's	0.87 [0.73, 0.94]	0.87 [0.74, 0.94]
(Patwardhan et al., 2006)	WordNet	n/a	0.91
(Alvarez and Lim, 2007)	WordNet	n/a	0.91
(Yang and Powers, 2005)	WordNet	0.87 [0.73, 0.91]	0.92 [0.84, 0.96]
(Hughes et al., 2007)	WordNet	0.90	n/a
Personalized PageRank	WordNet	0.89 [0.77, 0.94]	n/a
Bag of words	Web corpus	0.85 [0.70, 0.93]	0.84 [0.69, 0.93]
Context window	Web corpus	0.88 [0.76, 0.95]	0.89 [0.77, 0.95]
Syntactic contexts	Web corpus	0.76 [0.54, 0.88]	0.74 [0.51, 0.87]
SVM	Web, WN	0.92 [0.84, 0.96]	0.93 [0.85, 0.97]

Table 7: Comparison with previous approaches for MC.

not have a held-out set, so we used the standard settings of Weka, without trying to modify parameters, e.g. C. Each word pair is scored with the number of pairs that were considered to have less similarity using the SVM. The results using 10-fold cross-validation are shown in Table 6. A combination of all methods produces the best results reported so far for both datasets, statistically significant for RG.

6 Related work

Contrary to the WordSim353 dataset, common practice with the RG dataset has been to perform the evaluation with Pearson correlation. In our believe Pearson is less informative, as the Pearson correlation suffers much when the scores of two systems are not linearly correlated, something which happens often given due to the different nature of the techniques applied. Some authors, e.g. Alvarez and Lim (2007), use a non-linear function to map the system outputs into new values distributed more similarly to the values in the gold-standard. In their case, the mapping function was $\exp\left(\frac{-x}{4}\right)$, which was chosen empirically. Finding such a function is dependent on the dataset used, and involves an extra step in the similarity calculations. Alternatively, the Spearman correlation provides an evaluation metric that is independent of such data-dependent transformations.

Most similarity researchers have published their

Word pair	M&C	SVM	Word pair	M&C	SVM
automobile, car	3.92	62	crane, implement	1.68	26
journey, voyage	3.84	54	brother, lad	1.66	39
gem, jewel	3.84	61	car, journey	1.16	37
boy, lad	3.76	57	monk, oracle	1.1	32
coast, shore	3.7	53	food, rooster	0.89	3
asylum, madhouse	3.61	45	coast, hill	0.87	34
magician, wizard	3.5	49	forest, graveyard	0.84	27
midday, noon	3.42	61	monk, slave	0.55	17
furnace, stove	3.11	50	lad, wizard	0.42	13
food, fruit	3.08	47	coast, forest	0.42	18
bird, cock	3.05	46	cord, smile	0.13	5
bird, crane	2.97	38	glass, magician	0.11	10
implement, tool	2.95	55	rooster, voyage	0.08	1
brother, monk	2.82	42	noon, string	0.08	5

Table 8: Our best results for the MC dataset.

Method	Source	Spearman
(Strube and Ponzetto, 2006)	Wikipedia	0.19–0.48
(Jarmasz, 2003)	WordNet	0.33–0.35
(Jarmasz, 2003)	Roget’s	0.55
(Hughes and Ramage, 2007)	WordNet	0.55
(Finkelstein et al., 2002)	Web corpus, WN	0.56
(Gabrilovich and Markovitch, 2007)	ODP	0.65
(Gabrilovich and Markovitch, 2007)	Wikipedia	0.75
SVM	Web corpus, WN	0.78

Table 9: Comparison with previous work for WordSim353.

complete results on a smaller subset of the RG dataset containing 30 word pairs (Miller and Charles, 1991), usually referred to as MC, making it possible to compare different systems using different correlation. Table 7 shows the results of related work on MC that was available to us, including our own. For the authors that did not provide the detailed data we include only the Pearson correlation with no confidence intervals.

Among the unsupervised methods introduced in this paper, the context window produced the best reported Spearman correlation, although the 0.95 confidence intervals are too large to allow us to accept the hypothesis that it is better than all others methods. The supervised combination produces the best results reported so far. For the benefit of future research, our results for the MC subset are displayed in Table 8.

Comparison on the WordSim353 dataset is easier, as all researchers have used Spearman. The figures in Table 9) show that our WordNet-based method outperforms all previously published WordNet methods. We want to note that our WordNet-based method outperforms that of Hughes and Ramage (2007), which uses a similar method. Although there are some differences in the method, we think that the main performance gain comes from the use of the disambiguated glosses, which they did not use. Our distributional methods also outperform all

other corpus-based methods. The most similar approach to our distributional technique is Finkelstein et al. (2002), who combined distributional similarities from Web documents with a similarity from WordNet. Their results are probably worse due to the smaller data size (they used 270,000 documents) and the differences in the calculation of the similarities. The only method which outperforms our non-supervised methods is that of (Gabrilovich and Markovitch, 2007) when based on Wikipedia, probably because of the dense, manually distilled knowledge contained in Wikipedia. All in all, our supervised combination gets the best published results on this dataset.

7 Conclusions and future work

This paper has presented two state-of-the-art distributional and WordNet-based similarity measures, with a study of several parameters, including performance on similarity and relatedness data. We show that the use of disambiguated glosses allows for the best published results for WordNet-based systems on the WordSim353 dataset, mainly due to the better modeling of relatedness (as opposed to similarity). Distributional similarities have proven to be competitive when compared to knowledge-based methods, with context windows being better for similarity and bag of words for relatedness. Distributional similarity was effectively used to cover out-of-vocabulary items in the WordNet-based measure providing our best unsupervised results. The complementarity of our methods was exploited by a supervised learner, producing the best results so far for RG and WordSim353. Our results include confidence values, which, surprisingly, were not included in most previous work, and show that many results over RG and WordSim353 are indistinguishable. The algorithm for WordNet-base similarity and the necessary resources are publicly available⁸.

This work pioneers cross-lingual extension and evaluation of both distributional and WordNet-based measures. We have shown that closely aligned wordnets provide a natural and effective way to compute cross-lingual similarity with minor losses. A simple translation strategy also yields good results for distributional methods.

⁸<http://ixa2.si.ehu.es/ukb/>

References

- E. Agirre and A. Soroa. 2009. Personalizing pagerank for word sense disambiguation. In *Proc. of EACL 2009*, Athens, Greece.
- M.A. Alvarez and S.J. Lim. 2007. A Graph Modeling of Semantic Similarity between Words. *Proc. of the Conference on Semantic Computing*, pages 355–362.
- J. Atserias, L. Villarejo, G. Rigau, E. Agirre, J. Carroll, B. Magnini, and P. Vossen. 2004. The meaning multilingual central repository. In *Proc. of Global WordNet Conference*, Brno, Czech Republic.
- D. Bollegala, Matsuo Y., and M. Ishizuka. 2007. Measuring semantic similarity between words using web search engines. In *Proceedings of WWW'2007*.
- A. Budanitsky and G. Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.
- H. Chen, M. Lin, and Y. Wei. 2006. Novel association measures using web search with double checking. In *Proceedings of COCLING/ACL 2006*.
- J. Daude, L. Padro, and G. Rigau. 2000. Mapping WordNets using structural information. In *Proceedings of ACL'2000*, Hong Kong.
- C. Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database and Some of its Applications*. MIT Press, Cambridge, Mass.
- L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppim. 2002. Placing Search in Context: The Concept Revisited. *ACM Transactions on Information Systems*, 20(1):116–131.
- E. Gabrilovich and S. Markovitch. 2007. Computing Semantic Relatedness using Wikipedia-based Explicit Semantic Analysis. *Proc of IJCAI*, pages 6–12.
- T. H. Haveliwala. 2002. Topic-sensitive pagerank. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 517–526.
- T. Hughes and D. Ramage. 2007. Lexical semantic relatedness with random graph walks. In *Proceedings of EMNLP-CoNLL-2007*, pages 581–589.
- M. Jarmasz. 2003. Roget's Thesaurus as a lexical resource for Natural Language Processing.
- J.J. Jiang and D.W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of International Conference on Research in Computational Linguistics*, volume 33. Taiwan.
- C. Leacock and M. Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. *WordNet: An Electronic Lexical Database*, 49(2):265–283.
- D. Lin. 1998a. An information-theoretic definition of similarity. In *Proc. of ICML*, pages 296–304, Wisconsin, USA.
- D. Lin. 1998b. Automatic Retrieval and Clustering of Similar Words. In *Proceedings of ACL-98*.
- G.A. Miller and W.G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.
- J. Nivre. 2006. *Inductive Dependency Parsing*, volume 34 of *Text, Speech and Language Technology*. Springer.
- S. Pado and M. Lapata. 2007. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199.
- S. Patwardhan and T. Pedersen. 2006. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of the EACL Workshop on Making Sense of Sense: Bringing Computational Linguistics and Psycholinguistics Together*, pages 1–8, Trento, Italy.
- P. Resnik. 1995. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. *Proc. of IJCAI*, 14:448–453.
- H. Rubenstein and J.B. Goodenough. 1965. Contextual correlates of synonymy. *Communications of the ACM*, 8(10):627–633.
- M Ruiz-Casado, E. Alfonseca, and P. Castells. 2005. Using context-window overlapping in Synonym Discovery and Ontology Extension. In *Proceedings of RANLP-2005*, Borovets, Bulgaria,.
- M. Sahami and T.D. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. *Proc. of WWW*, pages 377–386.
- M. Strube and S.P. Ponzetto. 2006. WikiRelate! Computing Semantic Relatedness Using Wikipedia. In *Proceedings of the AAI-2006*, pages 1419–1424.
- P.D. Turney. 2001. Mining the Web for Synonyms: PMI-IR versus LSA on TOEFL. *Lecture Notes in Computer Science*, 2167:491–502.
- P. Vossen, editor. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publishers.
- Z. Wu and M. Palmer. 1994. Verb semantics and lexical selection. In *Proc. of ACL*, pages 133–138, Las Cruces, New Mexico.
- D. Yang and D.M.W. Powers. 2005. Measuring semantic similarity in the taxonomy of WordNet. *Proceedings of the Australasian conference on Computer Science*.