Comparing Different Properties Involved in Word Similarity Extraction*

Pablo Gamallo Otero[1]

Universidade de Santiago de Compostela, Galiza, Spain
pablo.gamallo@usc.es

Abstract. In this paper, we will analyze the behavior of several parameters, namely type of contexts, similarity measures, and word space models, in the task of word similarity extraction from large corpora. The main objective of the paper will be to describe experiments comparing different extraction systems based on all possible combinations of these parameters. Special attention will be paid to the comparison between syntax-based contexts and windowing techniques, binary similarity metrics and more elaborate coefficients, as well as baseline word space models and Singular Value Decomposition strategies. The evaluation leads us to conclude that the combination of syntax-based contexts, binary similarity metrics, and a baseline word space model makes the extraction much more precise than other combinations with more elaborate metrics and complex models.

1 Introduction

Most of the existing work on word similarity extraction has in common two properties: the observation that semantically related words will appear in similar contexts and the use of word space models built on the basis of such co-occurrence observations. Yet, the underlying methods can differ in four different aspects: their definition of context, the way they calculate similarity from the contexts each word appears in, the way they modify the word space model (singular value decomposition, association values, etc.), and finally, the algorithm used to perform pairwise word comparisons.

There are many interesting works comparing the accuracy of different approaches on word similarity extraction. However, most of them are focused only on one parameter of variation. Some compare systems on the basis of the type of context, namely window and syntactic-based methods [10]. Other compare several similarity measures [5]. Some are interested in testing whether changes in the word space model can improve the results [13]. And, there is also some work comparing the computational efficiency of the underlying algorithm [20].

The main contribution of this paper is to compare word similarity systems on the basis of several parameters or ranges of variation, and not only considering one of them as it was usual in the literature. For this purpose, four parameters will be taken

* This work has been funded by the Galician Government (*Consellaría de Industria e Innovación* and *Consellaría de Educación e Ordenación Universitaria*)

into account: types of contexts (C), similarity measures (S), strategies to build word space models (M), and algorithms to compute similarity between pairwise words (A). A system is defined as a tuple of 4 elements, (c, s, m, a), where c is a type of context, s a similarity measure, m a word space model, and a an algorithm. So, according to this range of variation, we will define the cartesian product of all possible 4-tuples:

$$C \times S \times M \times A =$$
$$\{(c, s, m, a) | c \in C \text{ and } s \in S \text{ and } m \in M \text{ and } a \in A\}$$

where each 4-tuple is an evaluable system. In this paper, we will define 3 contexts, 10 similarity coefficients, 3 word space models, and 1 algorithm. As all systems share the same algorithm, all comparisons will be made among the remaining 3 parameters. As far as we know, up to now, no work has attempted to compare more than two parameters of variation against the same corpus. So, the main contribution of this paper is to compare 65 different systems, built from much of all possible triplets (90) containing C, S, and M.

Another contribution of the paper is to describe a large-scale evaluation including a new kind of gold standard. In addition to WordNet [7], we will also use as reference for evaluation a closed terminology, namely a list of proper names annotated with three sharp categories: countries, capitals, and English towns. The use of such a closed list as gold standard tries to overcome some of the problems associated with standard thesaurus, namely the fact that an extraction system can compute many correct word pairs which are all counted as wrong since they are not in the thesaurus. With the use of a closed list of all countries, capitals, and English towns, this problem does not arise. For instance, given a word tagged as being a country, and given the most similar word extracted by the system, if it this word is not tagged as a country, it is sure that it is not a country. All words correctly proposed by the system must be in the gold standard and, therefore, will always be correctly evaluated.

The evaluation described in this paper will lead us to conclude that, on the one hand, the systems based on syntactic contexts tend to be better than the windowing techniques, and on the other, it is very difficult to perform better than the simplest metrics and the baseline word space models.

The remainder of the paper is organized as follows. Section 2 enumerates some works comparing different similarity extraction systems according to only one parameter. In Section 3, we describe the different parameters of variation that will be used in our experiments. And finally, in Section 4, we will introduce some corpus-based experiments, define the evaluation protocol and analyze the results performed by 65 systems against the same corpus (BNC).

2 Related Work

There are much previous work aimed to evaluate and compare different strategies to extract word similarity. Some compare the influence of different types of contexts. In [10], a syntax-based method is carefully compared to a windowing technique. The former is shown to perform better for high-frequency words, while the windowing method

is the better performer for low-frequency words. The experiments performed made use of very small text corpora, probably due to the low efficiency of the syntactic techniques available at that time. Similar experiments were performed more recently [16, 17, 21]. All of them state that syntax-based methods outperform windowing techniques thanks to a drastic reduction of noise.

Other works compare the performance of different similarity measures. However, no agreement has been achieved concerning the best coefficients. In [14], the best performance was reached by the metric defined by the author. In [5], the best one was a specific version of Dice, and in [2], the best results were obtained by the simplest metrics, namely those based on merely counting contexts with non-zero values (i.e., binary measures).

There exists a large family of experiments comparing standard word space models to models previously reduced by Singular Value Decomposition (SVD). In [13], the best results are achieved using SVD, combined with large word contexts defined at the level of the document. In [22], SVD is outperformed by a more basic word space model. However, in [19, 15, 3], SVD combined with small window-based contexts outperform other approaches. In all these experiments, the evaluation uses as gold standard popular tests as TOEFL where the system has to choose the most appropriate synonym for a given word given a restricted list of four candidates. To compare the accuracy of two (or more) methods, it is assumed that the system makes the right decision if the correct word is ranked highest among the four alternatives. The main drawback of such an evaluation derives from the size of the test itself. Each word is compared to only other three words, and not to many thousands as in more reliable large-scale evaluations.

In addition, there are other works comparing word space models with regard to the type of association value (or weight) defining word-by-context co-occurrences [5, 2]. Like in the case of similarity metrics, there is no agreement concerning the best weight function for word similarity extraction.

Finally, we also can find some work measuring both the complexity and computational efficiency of the algorithm implemented to make pairwise comparisons [9, 20]. As the accuracy of any extraction system does not depend on the chosen algorithm, we will not compare systems with regard to this specific parameter.

Unlike the studies sketched above which make comparisons according to one or in some cases two parameters of variation, in this paper, we will compare different extraction systems with regard to 3 parameters.

3 Systems and Range of Variation

As has been said above, a system to extract word similarity can be defined as a 4-tuple consisting of:

- a type of context,
- a similarity measure,
- a word space model defined as a word-by-context co-occurrence matrix,
- an algorithm to compare pairs of words in an efficient way.

3.1 Types of Contexts

The systems we will compare were implemented according to 3 different types of word contexts. Two types of windowing strategies and one syntax-based method. As far the windowing strategies are concerned, contexts can be defined using the immediately adjacent words, within a window of n words. Two different techniques can be applied: one defining contexts as bag of words, called *BOW*, and the other taking into account word order (*WO*). The technique based on bag of words builds context vectors considering simple words as dimensions, regardless of their positions within the window. By contrast, the *WO* technique uses word order to define context vectors, which is considered to be useful to simulate syntactic behavior. According to Rapp [18], this window technique is, then, closer to the syntax-based approach.

Our syntactic strategy (*SYN*) relies on dependency-based robust parsing. Dependencies are generated by means of *DepPattern*[1], a rule-based partial parser which can process 5 languages: English, Spanish, Galician, Portuguese, and French. The 5 grammars are very generic, they are constituted by about 20-30 rules each. To extract syntax-based contexts from dependencies, we used the co-compositional methodology defined in [8]. The *DepPattern* toolkit also includes a script aimed to extract co-compositional contexts from the dependencies generated by the parser.

3.2 Similarity Measures

The systems are built using 10 similarity coefficients, which represent much of the metrics defined in [14, 5, 2]. The simplest measures (suffix "Bin") transform all vectors into binary values: binary overlapping (OverBin), binary Dice (DiceBin), binary Jaccard (JaccBin), and binary Cosine (CosBin). By contrast, Cosine (Cos), Euclidian distance (Eucl), City-Block (City), Dice (DiceMin), and Jaccard (JaccMin) use vectors with co-occurrence (or weighted) values. The 10 similarity metrics between two words, w_1 and w_2, are defined in Table 1, where $BIN(w_1)$ stands for a set representation of the binary vector defining word w_1. This vector is the result of transforming the real-valued vector with co-occurrences or log-likelihood scores into a vector with binary values. The length $\parallel BIN(w_1) \parallel$ of a binary vector $BIN(w_1)$ is the number of non-zero values. On the other hand, $A(w_1, c_j)$ is an association value of a vector of length n, with j, i, and k ranging from 1 to n. In our experiments, the association value stands for either the simple co-occurrences of word w_1 with a contextual expression c_j, or a weight computed using the log-likelihood ratio between the word and its context. For Cosine, the association values of two words with the same context are joined using their product, while for JaccardMin [11, 12] and DiceMin [5, 23] only the smallest association weight is considered (in those works, they are noted as Jaccard† and Dice†, respectively). For the Lin coefficient, the association values of common contexts are summed [14], where $c_j \in C_{1,2}$ if and only if $A(w_1, c_j) > 0$ and $A(w_2, c_j) > 0$. Finally, in City, $|x - y|$ represents an absolute value. In sum, we use two types of similarity coefficients: those based on binary vectors (baseline metrics) and those relying on association values.

[1] *DepPattern* is a linguistic toolkit, with GPL licence, which is available at: `http://gramatica.usc.es/pln/tools/deppattern.html`

$$\mathsf{OverBin}(w_1, w_2) = \parallel \mathsf{BIN}(w_1) \cap \mathsf{BIN}(w_2) \parallel$$

$$\mathsf{DiceBin}(w_1, w_2) = \frac{2 \parallel \mathsf{BIN}(w_1) \cap \mathsf{BIN}(w_2) \parallel}{\parallel \mathsf{BIN}(w_1) \parallel + \parallel \mathsf{BIN}(w_2) \parallel}$$

$$\mathsf{JaccBin}(w_1, w_2) = \frac{\parallel \mathsf{BIN}(w_1) \cap \mathsf{BIN}(w_2) \parallel}{\parallel \mathsf{BIN}(w_1) \cup \mathsf{BIN}(w_2) \parallel}$$

$$\mathsf{CosBin}(w_1, w_2) = \frac{\parallel \mathsf{BIN}(w_1) \cap \mathsf{BIN}(w_2) \parallel}{\sqrt{\parallel \mathsf{BIN}(w_1) \parallel}\sqrt{\parallel \mathsf{BIN}(w_2) \parallel}}$$

$$\mathsf{City}(w_1, w_2) = \sum_j |\mathsf{A}(w_1, c_j) - \mathsf{A}(w_2, c_j)|$$

$$\mathsf{Eucl}(w_1, w_2) = \sqrt{\sum_j (\mathsf{A}(w_1, c_j) - \mathsf{A}(w_2, c_j))^2}$$

$$\mathsf{Cosine}(w_1, w_2) = \frac{\sum_j \mathsf{A}(w_1, c_j)\mathsf{A}(w_2, c_j)}{\sqrt{\sum_j (\mathsf{A}(w_1, c_j))^2}\sqrt{\sum_k (\mathsf{A}(w_2, c_k))^2}}$$

$$\mathsf{DiceMin}(w_1, w_2) = \frac{2\sum_j \min(\mathsf{A}(w_1, c_j), \mathsf{A}(w_2, c_j))}{\sum_j \mathsf{A}(w_1, c_j) + \sum_k \mathsf{A}(w_2, c_k)}$$

$$\mathsf{JaccardMin}(w_1, w_2) = \frac{\sum_j \min(\mathsf{A}(w_1, c_j), \mathsf{A}(w_2, c_j))}{\sum_j \max(\mathsf{A}(w_1, c_j), \mathsf{A}(w_2, c_j))}$$

$$\mathsf{Lin}(w_1, w_2) = \frac{\sum_{c_i \in C_{1,2}} (\mathsf{A}(w_1, c_j) + \mathsf{A}(w_2, c_j))}{\sum_j \mathsf{A}(w_1, c_j) + \sum_k \mathsf{A}(w_2, c_k)}$$

Table 1. 10 similarity measures

3.3 Word Space Models

In our experiments, we evaluate the performance of three different types of word space models. First, we call COOC the simplest method that takes as input a sparse matrix containing only word-by-context co-occurrences. This is the baseline model. No further operation was applied on the baseline matrix before computing word similarity.

The second method, called SVD, requires a dense matrix reduced by Singular Value Decomposition. Dimensionality reduction was performed with SVDLIBC.[2]. Before reduction, co-occurrence values were transformed into log-likelihood scores, as in most approaches to Latent Semantic Analysis [13].

The third method, called BORDAG, was defined in [2], and consists of the following tasks: all co-occurrences are weighted values (log-likelihood) and are ranked by decreasing significance. Then, only the N best ones are selected (where $N = 200$ in our experiments). This way, each word is associated, at most, with 200 non-zero weighted values. Given that corpus frequency follows the power-law distribution, only very frequent words co-occur with more than 200 other words. Even if such a filtering strategy only affects very frequent words, it allows us to reduce the number of pairwise comparisons (and thus runtime) significantly, while hopefully not decreasing accuracy with regard to the baseline model.

3.4 Algorithm

The naive algorithm to extract word similarity looks at each word and compares it with each other word, checking all contexts to see if they are shared. Complexity is quadratic. Yet, it is possible to make the algorithm simpler. Because of the power-law distribution of word-context co-occurrences, most word pairs have nothing in common. So, there is no reason to check them. Following [9, 20], we implemented an algorithm that only compares word pairs sharing at least one context. As the list of words sharing a context is small (in general, less than 1000), the quadratic complexity of the entire algorithm turns out to be manageable.

4 Experiments and Large-Scale Evaluation

Given the parameters of variation described in the last section and the cartesian product of all possible 4-tuples, we could evaluate $3 \times 10 \times 3 \times 1$ systems, that is, 90 different strategies to extract word similarity. However, because of the computational complexity derived from SVD reduction, we only combined this word space model with one type of context, namely BOW (Bag Of Words). Moreover, as the matrices reduced with SVD do not allow similarity computation with binary metrics, at the end, we evaluate just 65 extraction systems. For instance, SYN-DiceBin-COOC stands for a system constituted by a syntactic-based context (SYN), a binary dice metric (DiceBin), and a simple word-by-context coocurrence matrix (COOC). To simplify, the name of each system is not provided with the specific algorithm, since it will not be evaluated. We also can use names to refer to sets of systems. For instance, SYN-COOC represents all systems made

[2] `http://tedlab.mit.edu/~dr/svdlibc/`

of syntactic contexts and co-occurrences, while SYN represents the more abstract set containing all syntax-based systems.

4.1 Corpus and Gold Standards

The experiments were performed on the British National Corpus (BNC)[3] corpus, containing about 100 million word tokens. For evaluation, we selected the $15,000$ most frequent proper names, on the one hand, and the $10,000$ most frequent common nouns, on the other. These are the target words to be evaluated. Proper names are evaluated taking as gold standard a closed list of countries, world capitals, and English towns[4]. This list contains 1610 names, each with a specif tag. Some (very few) contain more than one tag. For instance, *London* is both a world capital and an English town. Let's note that here the similarity relation is quite narrow. It is restricted to the relation of direct co-hyponymy, e.g., *England* is similar to *China* because they are both countries. By contrast, *England* is not related to *London*. 749 out of 1610 terms of the list are among the $15,000$ most frequent proper names in the BNC corpus. With the use of a closed list of related terms, we are sure that all similar words correctly proposed by a system are in the gold standard and, then, are correctly evaluated. Using Wordnet, however, many similar words that were correctly proposed by the system may not be in the gold standard, and consenquently, may be incorrectly considered as wrong.

To evaluate the common nouns, we take as gold standard WordNet [7]. Here the notion of similarity is larger than in the previous gold standard. The set of similar words of a given word is constituted by all those related to it by any direct semantic relationship (synonymy, meronymy, hyperonymy, ...), and indirectly, by those co-hyponyms selected from its hyperonyms at the first level. $6,943$ common nouns from WordNet were found among the $10,000$ most frequent ones in the corpus.

Given the $15,000$ most frequent proper names and the three types of contexts defined above, we build three $15,000$-by-$15,000$ word-by-context co-occurrence matrices with proper names and contexts of proper names. Contexts are also the $15,000$ most frequent ones. They change according to the type of context selected to build the system. For instance, if the type of context is defined from syntactic dependencies, the matrix contains the $15,000$ most frequent syntactic contexts of proper names. So, the generated matrices are constituted by the same target words (the most frequent proper names), but they differ in the contexts: syntax-based, word order or bag of words. The same is done with the $10,000$ most frequent common nouns: we build three $10,000$-by-$10,000$ word-by-context co-occurrence matrices with common nouns and contexts of common nouns. All these matrices represent the baseline word space model (COOC), from which BORDAG and SVD are derived. Previous tests led us to select $15,000$-by-300 and $10,000$-by-300 as those reduced matrices giving the optimal results for SVD-based systems.

[3] `http://www.natcorp.ox.ac.uk/`
[4] `AUTHOR-URL`

4.2 Evaluation

To evaluate the quality of all tested extraction systems, we elaborate an automatic and large-scale evaluation protocol with the following characteristics. Each system provides for each target word (proper name or common noun of the input matrix), a ranked list with its top-10 most similar words. A similar word of the ranked list is considered a true positive if it is related in the gold standard to the target word. For instance, if *China* is in the top-10 ranked list of *England*, and both proper names are tagged with the same tag (country) in the gold standard, then *China* is counted as a true positive. To measure the quality of each system, we use "mean Average Precision" (mean-AP) [4]. Average Precision (AP) consists in evaluating the average quality of the ranking produced for each test word. More precisely, it is the average of the precision scores at the rank locations of each true positive. Assuming a word contains N similar words extracted by the system, in which K are true positives, and p_i the rank of i-th positive, AP is:

$$AP = \frac{1}{N}\sum_{i=1}^{k}\frac{i}{p_i}$$

Note that i/p_i is just the precision value at the i-th positive in this iterative process. Let's see an example. If 2 out 10 ranked words were found at ranking positions 2 and 5, the AP in percent in this case is: $1/10*(1/2+2/5)*100 = 9\%$. 100% is achieved when the 10 ranked words are related to the test word in the gold standard. Mean Average Precision is the sum of average precisions divided by the number of evaluable words (i.e., words occurring in both the gold standard and the training corpus):

$$\textit{mean-AP} = \frac{1}{n}\sum_{i=1}^{n} AP_i$$

where n, the number of evaluable words, is 749 in the case of proper names, and $6,943$ for common nouns.

4.3 Results

Tables 2 and 3 shows the mean-AP scores obtained for all systems using respectively the $15,000$ most frequent proper names and the $10,000$ most frequent common nouns. Each column represents a combination between a type of context and a word space model, while rows stands for the 10 similarity metrics introduced above in 3.2. The best score in Table 2 is 59.04%, achieved by the system SYN-OverBin-BORDAG. In Table 3 the best mean-AP value merely achieves 16.54%, obtained by SYN-CosineBin-BORDAG. Even if the two tables differ significantly in the scale of their values, most systems have a similar behavior across the two evaluations. The main exception corresponds to the SVD-based systems (BOW-SVD), which are the only systems whose mean-AP scores improve when they are evaluated using WordNet (Table 3).

METRIC	SYN-COOC	WO-COOC	BOW-COOC	SYN-BORDAG	WO-BORDAG	BOW-BORDAG	BOW-SVD
CityBlock	5.24	2.7	17.77	4.17	1.88	5.58	3.01
CosineBin	50.06	50.62	47.62	47.85	39.25	38.96	
Cosine	36.50	10.55	39.08	32.67	11.09	34.77	3.56
DiceBin	50.68	46.68	46.58	49.25	38.66	38.97	
DiceMin	47.55	18.31	43.40	45.77	15.54	43.69	2.88
Euclidean	16.92	7.63	17.93	18.73	6.91	18.99	2.96
JaccBin	50.68	46.68	46.58	49.25	38.66	38.97	
JaccMin	47.55	18.31	43.40	45.77	15.54	43.69	3.20
Lin	23.57	8.86	25.48	24.51	8.11	25.34	
OverBin	46.52	28.39	30.43	**59.04**	41.86	38.99	

Table 2. Mean-AP of Proper Names using as gold-standard a list of countries, capitals, and towns.

METRIC	SYN-COOC	WO-COOC	BOW-COOC	SYN-BORDAG	WO-BORDAG	BOW-BORDAG	BOW-SVD
CityBlock	2.53	0.56	3.78	0.90	0.33	1.45	3.79
CosineBin	15.18	11.50	8.74	**16.54**	3.74	12.83	
Cosine	7.86	1.26	11.32	6.99	1.4	10.98	7.00
DiceBin	12.97	10.14	8.11	16.22	3.74	12.83	
DiceMin	11.23	2.76	7.28	12.65	1.80	11.76	4.84
Euclidean	2.64	0.98	2.78	2.71	0.91	3.63	3.37
JaccBin	12.97	10.14	8.11	16.22	3.74	12.83	
JaccMin	11.23	2.76	7.28	12.65	1.80	11.76	5.76
Lin	5.88	2.71	6.29	5.68	1.27	10.61	
OverBin	5.97	4.07	4.32	16.42	3.69	12.83	

Table 3. Mean-AP of common nouns using WordNet as gold-standard.

4.4 Ranking of systems

To interpret the results, instead of using test of significance looking for statistically different and similar groups of systems, we prefer ranking them using the mean of the two evaluations. For this purpose, mean-AP values are first normalized. Table 4 shows a sample of the 65 systems ranked by the mean of the normalized values. Notice that the best systems in the ranked list are based on syntactic contexts, binary similarity metrics, and the BORDAG word space model. Surprisingly, the system with the best score uses the simplest similarity metric (OverBin), which merely counts the number of contexts shared by the compared words. Systems with window-based contexts and metrics with association values appear at the bottom of the list.

Rank	System	Mean
1	SYN-OverBin-BORDAG	0.99
2	SYN-DiceBin-BORDAG	0.90
3	SYN-JaccBin-BORDAG	0.90
4	SYN-CosineBin-BORDAG	0.90
5	SYN-CosineBin-COCC	0.88
6	SYN-JaccBin-COCC	0.82
7	SYN-DiceBin-COCC	0.82
8	WO-CosineBin-COCC	0.77
9	SYN-DiceMin-BORDAG	0.76
10	SYN-JaccMin-BORDAG	0.76
...	...	...
20	WO-JaccardBin-COCC	0.69
25	BOW-Cosine-BORDAG	0.62
30	BOW-Lin-BORDAG	0.53
35	BOW-Lin-COOC	0.43
40	WO-OverBin-COCC	0.35
45	BOW-Cosine-SVD	0.22
50	BOW-JaccardMin-SVD	0.18
55	WO-Cosine-BORDAG	0.11
60	WO-Euclidean-COCC	0.07
65	WO-CityBlock-BORDAG	0.02

Table 4. Ranking of systems.

It is also possible to rank separately the different parameters of variation underlying the evaluated systems. Table 5 shows the mean and variance of each metric. Given a metric, we compute the average score obtained across all systems based on this metric. From this point of view, the best metric is now CosineBin. Let's note that the four binary metrics are at the top of the ranked list. This is in accordance with the evaluation described by Bordag [2], but not with other related work, such as [5], where DiceMin was considered as the best coefficient. In [14], no binary metric was evaluated. We think, however, that the evaluation described by Curran and Moens [5] is not entirely reliable. In their work, equivalent metrics, like DiceMin and JaccMin or DiceBin and JaccBin, achieved very different precision scores. This is not in accordance with the fact that

Jaccard and Dice coefficients should tend to yield the same similarity performance for any word. The Dice and Jaccard measures are fully equivalent, i.e., there is a monotonic transformation between their scores [6]. Notice that in our evaluation this pair of metrics produces almost always the same scores. It follows that our results are close to those expected by the theory. As far as the standard deviation (σ) is concerned, the table also shows how it increases from the top to the bottom of the list. The best metrics are then more stable across the different systems since they behave in the same way regardless of the context or model being used. Finally, Euclidean and CityBlock distances are not suited at all to deal with word similarity extraction.

Table 5 also shows the ranking of contexts and models. Whereas syntax-based contexts (SYN) perform clearly better than the two types of window-based contexts, the difference between BORDAG and the baseline model (COCC) is very small. Even if the best systems are based on the BORDAG model, its high standard deviation makes it quite instable. In particular, when it is combined with contexts of type WO the performance decreases in a significant way. By contrast, the word space model based on simple co-occurrences is more regular and stable, as we can infer from its low standard deviation. Very far from the scores achieved by these two models, we find SVD. Latent information resulting of factorization by Singular Value Decomposition, such as high-order co-occurrences, do not help to improve the task of word similarity extraction.[5]

Metric	**Mean**	σ
CosineBin	0.72	0.07
DiceBin	0.70	0.09
JaccBin	0.70	0.09
OverBin	0.58	0.18
JaccMin	0.48	0.26
DiceMin	0.47	0.27
Cosine	0.39	0.37
Lin	0.31	0.47
Euclidean	0.16	0.70
CityBlock	0.08	0.80
Context	**Mean**	σ
SYN	0.60	0.48
BOW	0.46	0.96
WO	0.28	0.90
Model	**Mean**	σ
BORDAG	0.48	0.94
COCC	0.47	0.64
SVD	0.15	0.71

Table 5. Ranking of metrics, contexts, and models.

[5] To be sure that our SVD-based systems were well implemented, we made a comparison with the LSA strategy underlying Infomap (`http://infomap-nlp.sourceforge.net/`). We used as training corpus a small sample of proper names from BNC. There were no significant differences between the results achieved with our systems and those obtained with Infomap.

5 Conclusions

The main contribution of this paper is to compare 65 different systems to extract word similarity under controlled circumstances by means of a large-scale evaluation, and by taking as gold-standard, both WordNet and a large list of proper names classified in three semantic categories.

The results of the experiments leave no doubt that, at least, for the task at stake and for the most frequent words of a corpus, the simplest similarity coefficients, based on binary values, are much more precise than more complex metrics requiring association values. This is not far from the main conclusions drawn by Bordag [2] from different experiments. In addition, syntactic contexts perform better than those based on windowing techniques (with or without taking into account word order). This is also in accordance with most experiments comparing both types of contexts. Regarding the word space model, it seems that Bordag-based systems performs slightly better than those based on basic co-occurrences, but differences are actually very small. SVD-based models, however, are much less precise in their results. So, to compute word similarity, it turns out to be difficult to overcome those systems relying on baseline strategies, namely those using binary metrics and simple co-occurrence matrices. Only dependency-based information seems to be more precise than more basic contexts based on windowing techniques.

Given that the syntactic parser used in our experiments only was constituted by very few rules (about 20), there is still room for improvement. In future work, we will compare the efficiency of different sets of syntactic-based contexts by integrating them in baseline systems with basic metrics and basic word space models. A different strategy to improve results would be to explore other theoretical paradigms for modeling new types of contexts and different word spaces, such as the proposal described in [1].

References

1. Marco Baroni and Alessandro Lenci. Concepts and properties in word space. *Italian Journal of Linguistics*, 20(1), 2008.
2. Stefan Bordag. A comparison of co-occurrence and similarity measures as simulations of context. In *9th CICLing*, pages 52–63, 2008.
3. R. Budiu and P. Pirolli. Navigation in degree-of-interest trees. In *Advance Visual Interface Conference*, 2006.
4. Zhuoran Chen. Assessing sequence comparison methods with the average precision criterion. *Bioinformatics*, 19, 2003.
5. James R. Curran and Marc Moens. Improvements in automatic thesaurus extraction. In *ACL Workshop on Unsupervised Lexical Acquisition*, pages 59–66, Philadelphia, 2002.
6. Stefan Evert. *The Statistics of Word Cooccurrences: Word Pairs and Collocations*. PhD thesis, Institut fr maschinelle Sprachverarbeitung, University of Stuttgart, 2005.
7. C. Fellbaum. A semantic network of english: The mother of all wordnets. *Computer and the Humanities*, 32:209–220, 1998.
8. Pablo Gamallo, Alexandre Agustini, and Gabriel Lopes. Clustering syntactic positions with similar semantic requirements. *Computational Linguistics*, 31(1):107–146, 2005.
9. James Gorman and James R. Curran. Scaling distributional similarity to large corpora. In *44th annual meeting of the Association for Computational Linguistics*, pages 361–368, Sydney, Australia, 2006.

10. Gregory Grefenstette. Evaluation techniques for automatic semantic extraction: Comparing syntactic and window-based approaches. In *Workshop on Acquisition of Lexical Knowledge from Text SIGLEX/ACL*, Columbus, OH, 1993.
11. Gregory Grefenstette. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers, USA, 1994.
12. Hiroyuki Kaji and Toshiko Aizono. Extracting word correspondences from bilingual corpora based on word co-occurrence information. In *16th Conference on Computational Linguistics (Coling'96)*, pages 23–28, Copenhagen, Danmark, 1996.
13. T.K. Landauer and S.T. Dumais. A solution to plato's problem: The latent semantic analysis theory of acquision, induction and representation of knowledge. *Psychological Review*, 10(2):211–240, 1997.
14. Dekang Lin. Automatic retrieval and clustering of similar words. In *COLING-ACL'98*, Montreal, 1998.
15. I. Matveeva, G. Levow, A. Farahat, and C. Royer. Terms representation with generalized latent semantic analysis. In *RANLP-2005*, 2005.
16. Sebastian Padó and Mirella Lapata. Dependency-based construction of semantic space models. *Computational Linguistics*, 33(2):161–199, 2007.
17. Yves Peirsman, Kris Heylen, and Dirk Speelman. Finding semantically related words in dutch. co-occurrences versus syntactic contexts. In *CoSMO Workshop*, pages 9–16, Roskilde, Denmark, 2007.
18. Reinhard Rapp. Automatic identification of word translations from unrelated english and german corpora. In *ACL'99*, pages 519–526, 1999.
19. Reinhard Rapp. A freely available automatically generated thesaurus of related words. In *LREC-2004*, pages 395–398, Lisbon, Portugal, 2004.
20. Pavel Rychlý and Adam Kilgarriff. An efficient algorithm for building a distributional thesaurus (and other sketch engine developments). In *45th Annual Meeting of the Association for Computational Linguistics*, pages 41–44, Prague, Czech Republic, 2007.
21. Violeta Seretan and Eric Wehrli. Accurate collocation extraction using a multilingual parser. In *21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 953–960, 2006.
22. P. Turney. Mining the web for synonyms: PMI-IR versus LSA on TOEFL. In *12th European Conference of Machine Learning*, pages 491–502, 2001.
23. Lonneke van der Plas and Gosse Bouma. Syntactic contexts for finding semantically related words. In *Meeting of Computational Linguistics in the Netherlands (CLIN2004)*, 2004.