ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-336

Combining Bootstrapping and Feature Selection for Improving a Distributional Thesaurus

Olivier Ferret[1]

Abstract. Work about distributional thesauri has now widely shown that the relations in these thesauri are mainly reliable for high frequency words and for capturing semantic relatedness rather than strict semantic similarity. In this article, we propose a method for improving such a thesaurus through its re-balancing in favor of middle and low frequency words. This method is based on a bootstrapping mechanism: a set of positive and negative examples of semantically related words are selected in a unsupervised way from the results of the initial measure and used for training a supervised classifier. This classifier is then applied for reranking the initial semantic neighbors. We evaluate the interest of this reranking for a large set of English nouns with various frequencies.

1 Introduction

The work we present in this article focuses on the automatic building of a thesaurus from a corpus. This kind of thesaurus typically gives for each entry a list of semantically similar words, each word being associated with the score evaluating its semantic similarity with the entry. Following work such as [10], [16] or [5], a widespread way to tackle the problem of building such thesauri from corpora is to rely on a semantic similarity measure for extracting the semantic neighbors of the target entries of the thesaurus. Three main ways of implementing such measures can be distinguished. The first one relies on handcrafted resources in which semantic relations, which are generally paradigmatic relations in this case, are clearly identified. Work exploiting WordNet-like lexical networks for building semantic similarity measures such as [3] or [19] falls into this category. These measures typically rely on the hierarchical structure of these networks, based on hypernymy relations. The second approach to implement similarity measures makes use of a less structured source of knowledge about words: their textual descriptions. Definitions from classical dictionaries or *glosses* from WordNet are typical examples of such descriptions. WordNet's *glosses* were used to support Lesk-like measures in [1] and more recently, measures were also defined from Wikipedia or Wiktionaries [9]. The last option for building similarity measures is the corpus-based approach, based on a generalization of the distributional hypothesis [7]: each word is characterized by the set of contexts from a corpus in which it appears and the semantic similarity of two words is computed from the proportion of contexts they share. This perspective was first adopted by [10] and [16] and then, explored in details in [5], [22] or [13].

The problem of improving the results of the "classical" distributional approach as it can be found in [5] for instance was already tackled by some work. A significant part of these proposals focus on the weighting of the elements that are part of the contexts of words but some of them imply more radical changes. The use of dimensionality reduction techniques, such as Latent Semantic Analysis in [18] or the redefinition of the distributional approach in a Bayesian framework in [14], can be classified into this second category. The first one is represented by work such as [2], in which the weights of context elements are turned into ranks, or [24], followed and extended by [23], that proposes a bootstrapping method for modifying the weights of context elements according to the semantic neighbors found by an initial distributional similarity measure.

The work we present in this article shares with [24] the use of bootstrapping but adopts a different perspective: the "best" semantic neighbors are not directly used for adapting the weights of distributional context elements but for training, as in [12], a supervised machine learning classifier. We show that the resulting classifier can be used for reranking the semantic neighbors found by the initial measure and correcting some of its deficiencies for building a distributional thesaurus.

2 Building of an initial thesaurus

2.1 Defining a distributional similarity measure

The use of bootstrapping requires in our case the definition of a semantic similarity measure with state-of-the-art results in classical evaluations for this kind of measures, that is to say, TOEFL-like tests [15] or the extraction of semantic neighbors for building a thesaurus [5]. Although our target language is English, we chose to limit deliberately the level of the tools applied for preprocessing texts to part-of-speech tagging and lemmatization to make possible the transposition of our method to a large set of languages. This seems to be a reasonable compromise between the approach of [8], in which none normalization of words is done, and the more widespread use of syntactic parsers, as in [5]. More precisely, we used *TreeTagger* [20] for performing the linguistic preprocessing of the corpus we relied on for building our state-of-the-art similarity measure. This corpus was the AQUAINT-2 corpus, a middle-size corpus made of around 380 million words coming from news articles. For the extraction of distributional data and the characteristics of the measure, we selected, by relying on an extended TOEFL test proposed in [8], the following options for the parameters of a distributional similarity measure:

- distributional contexts made of graphical co-occurrents: co-occurrents collected in a fixed-size window centered on each occurrence in the corpus of the target word. These co-occurrents were restricted to content words, *i.e.* nouns, verbs and adjectives;
- size of the window = 3 (one word on the left and right sides of the target word), *i.e.* very short range co-occurrents;

[1] CEA, LIST, Vision and Content Engineering Laboratory, F-91191 Gif-sur-Yvette, France, email: olivier.ferret@cea.fr

Table 1. Evaluation of semantic neighbor extraction [initial]

freq.	ref.	#eval. words	#syn. / word	recall	R-prec.	MAP	P@1	P@5	P@10	P@100
	W	10,473	2.9	24.6	8.2	9.8	11.7	5.1	3.4	0.7
all	M	9,216	50.0	9.5	6.7	3.2	24.1	16.4	13.0	4.8
# 14,670	WM	12,243	38.7	9.8	7.7	5.6	22.5	14.1	10.8	3.8
	W	3,690	3.7	28.3	11.1	12.5	17.2	7.7	5.1	1.0
high	M	3,732	69.4	11.4	10.2	4.9	41.3	28.0	21.9	7.9
# 4,378	WM	4,164	63.2	11.5	11.0	6.5	41.3	26.8	20.8	7.3
	W	3,732	2.6	28.6	10.4	12.5	13.6	5.8	3.7	0.7
middle	M	3,306	41.3	9.3	6.5	3.1	18.7	13.1	10.4	3.8
# 5,175	WM	4,392	32.0	9.8	9.3	7.4	20.9	12.3	9.3	3.2
	W	3,051	2.3	11.9	2.1	3.3	2.6	1.2	0.9	0.3
low	M	2,178	30.1	2.8	1.2	0.5	2.5	1.5	1.5	0.9
# 5,117	WM	3,687	18.9	3.5	2.1	2.4	3.3	1.7	1.5	0.7

- lenient filtering of contexts: removal of co-occurrents with only one occurrence;
- weighting function of co-occurrents in contexts = *Pointwise Mutual Information* between the target word and the co-occurrent;
- similarity measure between contexts, for evaluating the semantic similarity of two words = *Cosine* measure.

2.2 Thesaurus building and evaluation

The building of our initial distributional thesaurus from the previously defined similarity measure was performed as in [16] or [5] by extracting the closest semantic neighbors of each of its entries. More precisely, the selected measure was computed between each entry and its possible neighbors. These neighbors were then ranked in the decreasing order of the values of this measure and the first N ($N = 100$ here) neighbors were kept as the semantic neighbors of the entry. Both entries and possible neighbors were made of the AQUAINT-2 nouns whose frequency was higher than 10.

Table 1 shows the results of the evaluation of this extraction process, achieved by comparing the selected semantic neighbors with two complementary reference resources: WordNet 3.0 synonyms [17] [W], which characterize a semantic similarity based on paradigmatic relations, and the Moby thesaurus [21] [M], which gathers a larger set of types of relations and is more representative of *semantic relatedness* as it was defined in the first section[2]. The fourth column of Table 1, which gives the average number of synonyms and similar words in our references for the AQUAINT-2 nouns, also illustrates the difference of these two resources in terms of richness. A fusion of the two resources was also considered [WM]. As our main objective is to evaluate the extracted semantic neighbors and not the ability of our measure to rebuild the reference resources, these resources were filtered to discard entries and synonyms that are not part of the AQUAINT-2 vocabulary (see the difference between the number of words in the first column and the number of evaluated words of the third column). In distributional approaches, the frequency of words related to the size of the corpus is an important factor. Hence, we give our results globally but also for three ranges of frequencies that split our vocabulary into roughly equal parts: *high* frequency nouns (frequency > 1000), *middle* frequency nouns ($100 <$ frequency ≤ 1000) and *low* frequency nouns ($10 <$ frequency ≤ 100). These results take the form of several measures and start at the fifth column by the proportion of the synonyms and similar words of our references that are found among the first 100 extracted neighbors of each noun. As these neighbors are ranked according to their similarity value with their target word, the evaluation measures can be taken from the Information Retrieval field by replacing documents with synonyms and queries with target words (see the four last columns of Table 1). The R-precision (R-prec.) is the precision after the first R neighbors were retrieved, R being the number of reference synonyms; the Mean Average Precision (MAP) is the average of the precision value after a reference synonym is found; precision at different cut-offs is given for the 1, 5, 10 and 100 first neighbors. All these values are given as percentages.

The results of Table 1 lead to three main observations. First, they have globally a low level, which justifies our will to improve them. This weakness concerns both the recall of synonyms and their rank among semantic neighbors. Second, the level of results heavily depends on the frequency range of target words: the best results are obtained for high frequency words while evaluation measures significantly decrease for words whose frequency is less than 100 occurrences. Finally, the characteristics of the reference resources has a significant impact on results. WordNet provides a restricted number of synonyms for each noun while the Moby thesaurus contains for each entry a large number of synonyms and similar words. As a consequence, the precisions at different cut-offs have a significantly higher value with Moby as reference than with WordNet as reference.

Though our thesaurus was built by relying on a similarity measure selected by a classical test (extended TOEFL test) for such measures, it is interesting to compare the resulting thesaurus with already existing similar resources. The distributional thesaurus of Lin [16][3] is probably the publicly available thesaurus that is the most comparable to ours. Table 2 shows the global results of the Lin thesaurus in our evaluation framework. This thesaurus globally outperforms ours but the difference between its results and ours are clearly explainable by two main factors. First, the Lin thesaurus was built from syntactical co-occurrences while ours was built from window-based co-occurrences. The use of syntactical co-occurrences is known [5, 13] to be a better option in this context but it also requires to have a syntactic parser, which is not always possible and justifies our choice. Second, the Lin thesaurus is biased towards high and middle frequency words, which has clearly a positive impact as shown by the analysis of ours results: while the number of evaluated entries against WM for high and middle frequency words is equal to 8,313 for the

[2] Although the Moby thesaurus also contains related words, we will often use the term *synonym* for referring to all the words associated to its entries.

[3] http://webdocs.cs.ualberta.ca/ lindek/Downloads/sim.tgz

Table 2. Evaluation of extracted semantic neighbors with data of [16]

test set	ref.	#eval. words	#syn. / word	recall	R-prec.	MAP	P@1	P@5	P@10	P@100
	W	8,433	3.0	30.8	12.7	14.4	17.9	8.3	5.3	0.9
all	M	7,961	53.6	12.4	10.3	5.4	37.8	26.5	20.8	6.7
# 11,197	WM	9,823	44.5	12.7	11.6	8.1	36.1	23.7	18.2	5.6

Lin thesaurus and to 8,556 for ours, it is equal to 5,117 for low frequency entries in our case while it is only equal to 1,510 in the case of Lin. This bias can also be observed by the fact that the average number of synonyms and related words for each evaluated entry is higher for the Lin thesaurus than for ours.

3 Improving a distributional thesaurus

3.1 Principles

The analysis of the results of our state-of-the-art distributional similarity measure has showed that such measure has good results for some words and rather poor results for others. This is *a priori* an interesting configuration for applying bootstrapping as we can expect relying on "good" words for improving the similarity measure for the other words. [24] had already used bootstrapping in a context close to ours, the acquisition of relations of textual entailment between words, but our preliminary experiments for transposing this approach to our problem were not successful and led to a global decrease of results of 25% on average for the synonyms of WordNet and 11% for the related words of Moby [6]. Instead of using the results of an initial similarity measure for modifying directly the weights of elements in distributional contexts, we adopted a more indirect approach, based on the work of [12].

[12] demonstrated that it is possible to train and to apply with a good level of results a supervised statistical classifier, more precisely a Support Vector Machine (SVM) classifier, for deciding whether two words are synonyms or not. The term *synonym* must be taken here with caution as the gold standard that was used for evaluating the classifier is more suitable for testing *semantic relatedness* than *semantic similarity* in its strict sense. This work also showed that the value of the decision function of the SVM classifier, only used for its sign in a binary classification task, can play the same role as a similarity measure such as the one defined in the previous section for ranking the semantic neighbors of a word.

In our context, we do not have a set of manually annotated positive and negative examples for training such classifier. However, the similarity measure of Section 2 can be exploited for building such examples. This measure is a means for evaluating the semantic similarity of two words but is not able to discriminate directly words that are actually semantically similar from others[4]. Nevertheless, it can be used more indirectly for selecting a set of positive and negative examples in an unsupervised way while minimizing the number of errors in performing this task, that is to say, examples taken as positive whereas they are actually negative and examples taken as negative whereas they are actually positive. Following this perspective, it is possible to train a SVM classifier with this set of examples and to apply it for reranking the semantic neighbors given by our initial measure. The overall approach can be summarized as follows:

- building of a distributional similarity measure from a corpus;
- application of this measure for finding semantic neighbors;
- unsupervised selection of positive and negative examples of semantically similar words from the base of semantic neighbors;
- training of a supervised machine learning classifier from the set of selected examples;
- application of the trained classifier for reranking the semantic neighbors found by the initial measure.

The key point for making this approach successful is the capacity to select in an unsupervised way a number of "good" positive and negative examples that is large enough for compensating for the errors that are inherent in such selection.

3.2 Representation of examples

Before presenting this key point in details, it is necessary to define precisely the nature of our examples and how they are represented. As we follow [12] for building our similar word classifier, our representation of examples is also taken from [12]: a positive example is made of a pair of words that are synonyms or more generally semantically similar; a negative example is made of a pair of words that are not semantically similar. The representation of such pairs of words for a SVM classifier is built by associating their distributional representations. This association is performed for each pair of words (w_1, w_2) by summing the weights of the co-occurrents shared by the distributional representations of the two words. Co-occurrents of w_x that are not co-occurrents of w_y are given a null weight. Hence, the representation of each example has the same form as the distributional representation of a word, *i.e.* a vector of weighted words.

3.3 Positive and negative example selection

Results of Table 1 lead to an obvious conclusion: finding positive examples is far more difficult than finding negative examples as the number of semantic neighbors of a thesaurus entry that are actually semantically linked to this word quickly decreases as the rank of these neighbors increase. In the experiments of Section 4, we built negative examples from positive examples by turning each positive example (A,B) into two negative examples: (A, *rank 10 neighbor of A*) and (B, *rank 10 neighbor of B*). Choosing neighbors with a higher rank would guarantee fewer false negative examples and in principle, better results. In practice, taking neighbors with a rather small rank for building negative examples is a better option, probably because these examples are more useful in terms of discrimination as they are close to the transition area between negative and positive examples. We also found experimentally that the strongly imbalanced value of the ratio between the number of negative and positive examples in [12], equal to 6.5, didn't give in our situation significantly higher results for reranking semantic neighbors than our value for this ratio, equal to 2.

For the selection of positive examples, Table 1 shows that a true semantic neighbor is more likely to be found when the thesaurus entry is a high frequency noun and the considered neighbor has a low

[4] Setting a threshold for performing such discrimination leads to poor results. The variability of similarity values across words also justifies our decision to use SVM in a classification mode rather than in a regression mode.

Table 3. Impact of the reranking of semantic neighbors [rerank-s]

freq.	ref.	R-prec.		MAP		P@1		P@5		P@10	
all	W	7.8	(-0.4)	9.4	(-0.4)	11.2	(-0.5) ‡	5.0	(-0.1) ‡	3.3	(-0.1) ‡
	M	7.1	(0.4)	3.4	(0.2)	27.3	(3.2)	17.6	(1.2)	13.7	(0.7)
	WM	8.0	(0.3)	5.7	(0.1)	24.6	(2.1)	14.9	(0.8)	11.4	(0.6)
high	W	9.6	(-1.5)	11.1	(-1.4)	15.2	(-2.0)	7.0	(-0.7)	4.7	(-0.4)
	M	9.9	(-0.3)	4.6	(-0.3)	39.8	(-1.5)	26.2	(-1.8)	20.7	(-1.2)
	WM	10.4	(-0.6)	5.9	(-0.6)	39.2	(-2.1)	24.9	(-1.9)	19.5	(-1.3)
middle	W	9.0	(-1.4)	11.2	(-1.3)	12.2	(-1.4)	5.4	(-0.4)	3.5	(-0.2)
	M	7.3	(0.8)	3.6	(0.5)	26.1	(7.4)	16.4	(3.3)	12.6	(2.2)
	WM	9.2	(-0.1)	7.2	(-0.2)	24.9	(4.0)	14.5	(2.2)	10.9	(1.6)
low	W	4.1	(2.0)	5.2	(1.9)	5.0	(2.4)	2.2	(1.0)	1.4	(0.5)
	M	2.1	(0.9)	1.0	(0.5)	7.6	(5.1)	4.5	(3.0)	3.5	(2.0)
	WM	3.7	(1.6)	3.7	(1.3)	7.6	(4.3)	3.9	(2.2)	2.9	(1.4)
training # 1,592	W	18.7	(-3.7)	21.0	(-3.1)	31.0	(-5.5)	10.9	(-1.1)	6.7	(-0.5)
	M	10.9	(-0.4)	5.9	(-0.4)	54.8	(-2.9)	30.4	(-1.2)	22.6	(-1.0)
	WM	14.7	(-1.3)	11.1	(-1.2)	56.0	(-4.5)	28.4	(-1.5)	20.6	(-1.0)

rank. Following these observations would have led us to select as positive examples all pairs of words (*high frequency thesaurus entry*, *first neighbor of the entry*). This option would have meant having a large number of examples – 4,378 positives examples – but with a significant error rate equal to 58.7% in the most favorable case (WM as gold standard with 4,164 evaluated words among the 4,378). Moreover, limiting the number of examples is interesting if it does not decrease results as it makes training quicker, which is particularly useful for optimizing parameters, and often speeds up the application of the resulting SVM classifier as it tends to reduce the number of support vectors of its model. Hence, we have proposed a more selective method for choosing positive examples among high frequency nouns. This method, illustrated by Figure 1, is based on the assumption that semantic similarity relations are symmetric, as for WordNet's synonyms. As a consequence, we have hypothesized that if a thesaurus entry A has as first neighbor a word B, this neighbor is more likely to be semantically linked to A if A is the first neighbor of B as a thesaurus entry. In practice, this kind of symmetry between thesaurus entries and rank 1 neighbors is found for 1,052 target words, which produces 526 positive examples, as pairs (A,B) and (B,A) correspond to the same examples. As expected, this selection scheme reduces the number of positive examples but also reduces the number of false positive examples, with an error rate for this set equal to 40.2% in the most favorable case.

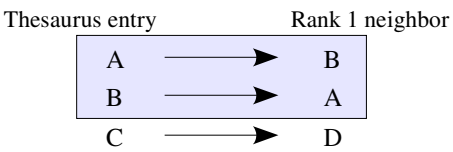

Figure 1. Selection of positive examples by relying on "symmetric" relations (A ↔ B)

Our 526 positive examples represent a small set compared to the 2,148 positive examples of [12]. However, this set can be extended by noting that high frequency nouns are actually not limited to nouns whose frequency is higher than 1000. By looking at the decrease of results according to decreasing word frequencies, it appears that an inflection point occurs a little bit before a frequency value corresponding to the median of the size of our vocabulary. As a consequence, we extended the set of our high frequency nouns to the first half (according to the decreasing values of their frequency) of the nouns of our vocabulary, which represents 7,335 nouns with a minimal frequency equal to 249. From these 7,335 nouns, 796 positive examples were selected according the principle presented above with an error rate equal to 40.3% in the most favorable case. The negative examples for these 796 positive examples were chosen as mentioned above, by taking pairs of words (*thesaurus entry*, *rank 10 neighbor of the entry*), but in this case for the $2 \times 796 = 1{,}592$ "symmetric" target words as this symmetry does not stand for rank 10 neighbors.

4 Experiments and evaluation

4.1 Implementation

The implementation of our reranking method requires to fix a set of parameters related to the SVM classifier. Similarly to [12], we adopted the RBF kernel and a grid search strategy for optimizing both the γ parameter of this kernel (*i.e.* the width of its Gaussian) and the C regularization parameter of SVM. This optimization was done by applying a 5-fold cross validation procedure to our set of 796 + 1,592 examples and taking the precision measure as evaluation function. The SVM model was built by using LIBSVM [4] and then applied to the 14,670 target nouns of our initial evaluation. More precisely, for each of these target nouns TN, the representation as an example of the word pair (TN, *neighbor*) was built for each of the 100 neighbors of this noun and submitted to the SVM model in classification mode. Finally, all the neighbors of TN were reranked according to the value of the decision function computed for each neighbor by the SVM model.

4.2 Evaluation

Table 3 gives the results of this reranking according to the same evaluation principles as in Section 2.2. The value of each measure comes with its difference with the corresponding value for the [initial] thesaurus in Table 1. Moreover, as this evaluation concerns a reranking process, the recall measure and the precision for the highest rank do not change and are not given again. The general trend is clear: the reranking process leads to a significant increase of results

Table 4. Results of the reranking of semantic neighbors after the filtering of their context [rerank-f]

ref.	R-prec.		MAP		P@1		P@5		P@10		P@100	
W	7.6	(-0.6)	8.7	(-1.1)	11.9	(0.2)	4.3	(-0.8)	2.7	(-0.7)	0.6	(-0.1)
M	6.3	(-0.4)	3.0	(-0.2)	25.7	(1.6)	15.7	(-0.7)	11.9	(-1.1)	4.5	(-0.3)
WM	7.3	(-0.4)	5.2	(-0.4)	23.7	(1.2)	13.2	(-0.9)	9.9	(-0.9)	3.6	(-0.2)

Table 5. Impact of feature selection for selecting training positive examples [rerank-f_s]

freq.	ref.	R-prec.		MAP		P@1		P@5		P@10	
all	W	8.1	(-0.1) ‡	9.7	(-0.1) ‡	11.7	(0.0) ‡	5.3	(0.2)	3.5	(0.1) ‡
	M	7.2	(0.5)	3.5	(0.3)	27.9	(3.8)	17.9	(1.5)	13.9	(0.9)
	WM	8.0	(0.3)	5.8	(0.2)	25.1	(2.6)	15.2	(1.1)	11.6	(0.8)
high	W	10.4	(-0.7)	11.8	(-0.7)	16.1	(-1.1)	7.7	(0.0) ‡	5.1	(0.0) ‡
	M	10.1	(-0.1) ‡	4.7	(-0.2)	40.8	(-0.5) ‡	27.1	(-0.9)	21.2	(-0.7)
	WM	10.6	(-0.4)	6.2	(-0.3)	40.2	(-1.1) ‡	26.0	(-0.8)	20.1	(-0.7)
middle	W	9.5	(-0.9)	11.7	(-0.8)	13.3	(-0.3) ‡	5.6	(-0.2) ‡	3.6	(-0.1) ‡
	M	7.3	(0.8)	3.6	(0.5)	27.2	(8.5)	16.3	(3.2)	12.5	(2.1)
	WM	9.4	(0.1)	7.4	(0.0)	26.2	(5.3)	14.5	(2.2)	10.8	(1.5)
low	W	3.5	(1.4)	4.7	(1.4)	4.5	(1.9)	2.1	(0.9)	1.4	(0.5)
	M	2.1	(0.9)	1.0	(0.5)	6.8	(4.3)	4.5	(3.0)	3.4	(1.9)
	WM	3.4	(1.3)	3.5	(1.1)	6.8	(3.5)	3.9	(2.2)	2.8	(1.3)
training # 2,980	W	13.8	(-2.0)	15.9	(-1.9)	21.2	(-3.3)	9.2	(-0.5)	5.9	(-0.2)
	M	10.5	(0.0)	5.4	(-0.1)	45.2	(-2.6)	28.3	(-0.3)	21.5	(-0.3)
	WM	12.3	(-0.7)	8.4	(-0.8)	44.9	(-3.8)	26.7	(-0.5)	19.9	(-0.4)

at the global scale (*all*) for gold standards M and WM. On the other hand, a decrease of results is noted for gold standard W[5]. In other words, compared to the initial similarity measure, this reranking process tends to favor similar words to the detriment of synonyms. This tendency is not surprising considering the principle of our reranking: similar words are more numerous than synonyms among the selected examples because the selection process does not correct a preexisting global bias towards similar words. The SVM model only accentuates this situation. The analysis of these results in terms of word frequency shows a second trend: the improvement due to the reranking process is all the more high since the frequency of the target noun is low. For low frequency nouns, this improvement is observed for all gold standards. Middle frequency nouns follow the general trend while results for high frequency nouns decrease for all gold standards. This observation means that the reranking process tends to make the initial similarity measure more balanced in relation to word frequency. Finally, the *training* row of Table 3 gives the results of our reranking procedure on the set of entries that were used for training our classifier. As expected, the values of the evaluation measures decrease compared to the [initial] thesaurus but the difference is not too high and has the same bias towards similar words: around 12.1% on average for WordNet as reference and 4.6% for Moby.

5 Feature selection for a better example selection

Our reranking process strongly depends on the relevance of the first neighbor of the target words used for the selection of examples. As a consequence, improving this selection means increasing the precision at rank 1 of our similarity measure. Focusing on such improvement can degrade results at a more global scale. However, this is not a problem in a bootstrapping approach as the biased measure is only used as a means for selecting the training examples of the SVM classifier. We built such biased measure by applying the idea of [24], but more radically: instead of using the results of the initial similarity measure for modifying the weights of co-occurrents in distributional contexts, we used them for filtering the content of these contexts since most elements in such contexts can be discarded for finding synonyms according to [11]. This filtering consists more precisely in removing all co-occurrents that are not part of the intersection between the context of the target word and the context of its first neighbor. It relies on the hypothesis that if the initial similarity measure is good enough, the features shared by the contexts of a target word and its first neighbor are more generally representative of the semantic similarity of a word with this target word. This unsupervised feature selection procedure was applied to the neighbors of all the entries of the [initial] thesaurus. These neighbors were then reranked by applying the *Cosine* measure to the filtered contexts. Table 4 shows the global results of this reranking, compared to the results of the [initial] thesaurus, and more particularly the fact that $P@1$ is the only measure improved by this procedure.

The resulting thesaurus, [rerank-f], was then used for selecting the training examples for the SVM classifier following the principles of Section 3.3. This led us to select 1,490 positive examples (and 2,980 negative examples) with 50.7% as best error rate, which constitutes a rather different configuration in comparison with Section 3.3: a higher number of positive examples but also a larger proportion of erroneous examples. The impact of this difference is illustrated by Table 5, which shows the results of the application of the SVM classifier with this new training set to the [initial] thesaurus. Hence, differences in this table are computed between the evaluation measures for this classifier and for the [initial] thesaurus of Table 1. This new

[5] The statistical significance of differences was evaluated by applying a paired Wilcoxon test with p-value < 0.05. Differences are non significant for values with the ‡ sign.

training set has clearly a positive influence on the global results of the SVM classifier as all positive differences at the global scale (*all*) are found statistically significant while all negative differences are found non significant. This influence can also be observed for the entries of the training set as the average decrease of their results is limited to 9% for WordNet as reference and 2% with Moby. Finally, Table 5 shows that this reranking is more favorable to the synonyms of high frequency nouns than the [rerank-s] reranking. As a consequence, the resulting thesaurus, [rerank-f_s], represents an interesting compromise: its results for high frequency nouns are quite similar to the results of the [initial] thesaurus while it significantly outperforms the [initial] thesaurus for less frequent nouns.

Table 6. Impact of our reranking procedures for the entry *inundation*

WordNet	torrent, deluge, flood
Moby	aspersion, bath, burial, cataclysm, deluge, dip, duck, exaggeration, excess, extravagance, flood, immersion ...+ 22 related words more
initial	prematurity, suport, closeup, late-spring, flooding, authoritarian, swallowtail, intussusception, fetishism, tri-state, bromate, slough ...
rerank-s	flooding, **flood**, avalanche, remoteness, blackout, congestion, mudslide, drought, rainfall, snowfall, downpour, **deluge** ...
rerank-f_s	**deluge**, flooding, **flood**, **torrent**, avalanche, salinity, settling, bromate, cyclone, blackout, quicklime, sunburn ...

Table 6 illustrates more precisely the impact of our two reranking procedures on the entry *inundation*, which is part of *middle* frequency nouns. In this table, the **WordNet** row gives all the reference synonyms for this entry in WordNet while the Moby row gives the first reference related words for this entry in Moby. In the [initial] thesaurus, the first neighbor of *inundation* that is present in one of our two reference resources (actually both of them) is the word *deluge*, at the 22^{th} rank. The [rerank-s] reranking improves this situation as the second neighbor in this thesaurus is both part of WordNet and Moby while the first neighbor of *inundation* is one of its synonyms that could be present in WordNet and Moby. Moreover, *deluge* appears as the 12^{th} neighbor of the *inundation* entry. Finally, the [rerank-f_s] reranking puts words that are semantically similar to the word *inundation* at the first four ranks: *deluge*, *flood* and *torrent* cover all synonyms of WordNet for this entry and are also present in Moby and *flooding* is a synonym of *inundation* already selected by the [rerank-s] reranking.

6 Conclusion and perspectives

In this article, we have presented a method based on bootstrapping for improving a distributional thesaurus. More precisely, this method relies on the reranking by the means of a SVM classifier of the semantic neighbors computed by applying an initial similarity measure. This classifier is trained from a set of positive and negative examples selected in an unsupervised way from the results of the initial similarity measure. The improvements brought by this method are more particularly noticeable for middle and low frequency words and in the case of similar words rather than for strict synonyms. We have also shown how to use feature selection for improving the selection of training examples for the SVM classifier. We plan to explore this problem further by studying more generally how unsupervised feature selection can be used to improve the building of distributional thesauri, especially for extracting synonyms.

REFERENCES

[1] Satanjeev Bano Banerjee and Ted Pedersen, 'Extended gloss overlaps as a measure of semantic relatedness', in *18^{th} International Conference on Artificial Intelligence (IJCAI-03)*, Acapulco, Mexico, (2003).

[2] Bartosz Broda, Maciej Piasecki, and Stan Szpakowicz, 'Rank-Based Transformation in Measuring Semantic Relatedness', in *22^{nd} Canadian Conference on Artificial Intelligence*, pp. 187–190, (2009).

[3] Alexander Budanitsky and Graeme Hirst, 'Evaluating wordnet-based measures of lexical semantic relatedness', *Computational Linguistics*, **32**(1), 13–47, (2006).

[4] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for support vector machines. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`, 2001.

[5] James R. Curran and Marc Moens, 'Improvements in automatic thesaurus extraction', in *Workshop of the ACL Special Interest Group on the Lexicon (SIGLEX)*, pp. 59–66, Philadelphia, USA, (2002).

[6] Olivier Ferret, 'Similarité sémantique et extraction de synonymes à partir de corpus', in *$17^{ème}$ Conférence sur le Traitement Automatique des Langues Naturelles (TALN 2010)*, Montréal, Canada, (2010).

[7] John R. Firth, *Studies in Linguistic Analysis*, chapter A synopsis of linguistic theory 1930-1955, 1–32, Blackwell, Oxford, 1957.

[8] Dayne Freitag, Matthias Blume, John Byrnes, Edmond Chow, Sadik Kapadia, Richard Rohwer, and Zhiqiang Wang, 'New experiments in distributional representations of synonymy', in *Ninth Conference on Computational Natural Language Learning (CoNLL)*, pp. 25–32, Ann Arbor, Michigan, USA, (2005).

[9] Evgeniy Gabrilovich and Shaul Markovitch, 'Computing semantic relatedness using wikipedia-based explicit semantic analysis', in *20^{th} International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pp. 6–12, (2007).

[10] Gregory Grefenstette, *Explorations in automatic thesaurus discovery*, Kluwer Academic Publishers, 1994.

[11] Masato Hagiwara, Yasuhiro Ogawa, and Katsuhiko Toyama, 'Context feature selection for distributional similarity', in *IJCNLP 2008*, pp. 553–560, Hyderabad, India, (2008).

[12] Masato Hagiwara, Yasuhiro Ogawa, and Katsuhiko Toyama, 'Supervised synonym acquisition using distributional features and syntactic patterns', *Information and Media Technologies*, **4**(2), 59–83, (2009).

[13] Kris Heylen, Yves Peirsmany, Dirk Geeraerts, and Dirk Speelman, 'Modelling Word Similarity: An Evaluation of Automatic Synonymy Extraction Algorithms', in *Sixth International Conference on Language Resources and Evaluation (LREC 2008)*, Marrakech, Morocco, (2008).

[14] Jun'ichi Kazama, Stijn De Saeger, Kow Kuroda, Masaki Murata, and Kentaro Torisawa, 'A bayesian method for robust estimation of distributional similarities', in *48^{th} Annual Meeting of the Association for Computational Linguistics*, pp. 247–256, Uppsala, Sweden, (2010).

[15] Thomas K. Landauer and Susan T. Dumais, 'A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge', *Psychological review*, **104**(2), 211–240, (1997).

[16] Dekang Lin, 'Automatic retrieval and clustering of similar words', in *ACL-COLING'98*, pp. 768–774, Montréal, Canada, (1998).

[17] George A. Miller, 'WordNet: An On-Line Lexical Database', *International Journal of Lexicography*, **3**(4), (1990).

[18] Sebastian Padó and Mirella Lapata, 'Dependency-based construction of semantic space models', *Computational Linguistics*, **33**(2), 161–199, (2007).

[19] Ted Pedersen, Siddharth Patwardhan, and Jason Michelizzi, 'Wordnet::similarity - measuring the relatedness of concepts', in *HLT-NAACL 2004, demonstrations*, pp. 38–41, Boston, Massachusetts, USA, (2004).

[20] Helmut Schmid, 'Probabilistic part-of-speech tagging using decision trees', in *International Conference on New Methods in Language Processing*, (1994).

[21] Grady Ward. Moby thesaurus. Moby Project, 1996.

[22] Julie Weeds, *Measures and Applications of Lexical Distributional Similarity*, Ph.D. dissertation, University of Sussex, 2003.

[23] Kazuhide Yamamoto and Takeshi Asakura, 'Even unassociated features can improve lexical distributional similarity', in *Second Workshop on NLP Challenges in the Information Explosion Era (NLPIX 2010)*, pp. 32–39, Beijing, China, (2010).

[24] Maayan Zhitomirsky-Geffet and Ido Dagan, 'Bootstrapping Distributional Feature Vector Quality', *Computational Linguistics*, **35**(3), 435–461, (2009).