Turning Distributional Thesauri into Word Vectors for Synonym Extraction and Expansion

Olivier Ferret
CEA, LIST, Vision and Content Engineering Laboratory,
Gif-sur-Yvette, F-91191 France.
olivier.ferret@cea.fr

Abstract

In this article, we propose to investigate a new problem consisting in turning a distributional thesaurus into dense word vectors. We propose more precisely a method for performing such task by associating graph embedding and distributed representation adaptation. We have applied and evaluated it for English nouns at a large scale about its ability to retrieve synonyms. In this context, we have also illustrated the interest of the developed method for three different tasks: the improvement of already existing word embeddings, the fusion of heterogeneous representations and the expansion of synsets.

1 Introduction

Early work about distributional semantics (Grefenstette, 1994; Lin, 1998; Curran and Moens, 2002) was strongly focused on the notion of distributional thesaurus. Recent work in this domain has been more concerned by the notions of semantic similarity and relatedness (Budanitsky and Hirst, 2006) and by the representation of distributional data. This trend has been strengthened even more recently with all work about distributed word representations and embeddings, whether they are built by neural networks (Mikolov et al., 2013) or not (Pennington et al., 2014).

From a more global perspective, distributional thesauri and distributional data, *i.e.* distributional contexts of words, can be considered as dual representations of the same semantic similarity information. Distributional data are an intensional form of this information that can take an extensional form as distributional thesauri by applying a similarity measure to them. Going from an intensional to an extensional representation corresponds to the rather classical process underlying the building of distributional thesauri. In the context of word embeddings, Perozzi et al. (2014a) extend this process to the building of lexical networks.

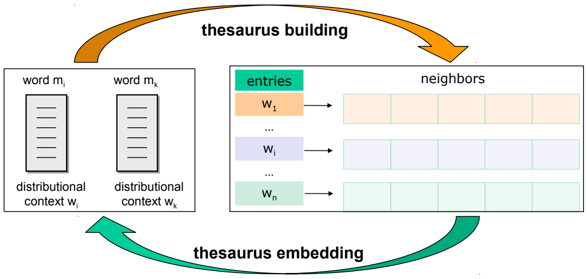

Figure 1: Duality of semantic information

Going to the other way, from an extensional to an intensional representation, is, as far as we know, a new problem in the context of distributional semantics. The interest of this transformation is twofold. First, whatever the initial form of the semantic knowledge, it can be turned into the most suitable form for a particular use. For instance, thesauri are more suitable for tasks like query expansion while word embeddings are more adapted as features for statistical classifiers. Second, each form is also associated with specific methods of improvement. A lot of work has been done for improving distributional contexts by studying various parameters, which has led to an important improvement of distributional thesauri. Conversely, work such as (Claveau et al., 2014) has focused on methods for improving thesauri themselves. It would clearly be interesting to transpose the improvements obtained in such a way to distributional contexts, as illustrated by Figure 1.

Hence, we propose in this article to investigate the problem of turning a distributional thesaurus into word embeddings, that is to say embedding a thesaurus. We will show that such process can

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 273–283,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

be achieved without losing too much information and moreover, that its underlying principles can be used for improving already existing word embeddings. Finally, we will illustrate the interest of such process for building word embeddings integrating external knowledge more efficiently and extending this knowledge.

2 Embedding Distributional Thesauri

A distributional thesaurus is generally viewed as a set of entries with, for each entry, a list of semantic neighbors ranked in descending order of semantic similarity with this entry. Since the neighbors of an entry are also entries of the thesaurus, such thesaurus can be considered as a graph in which vertices are words and edges are the semantic neighborhood relations between them, weighted according to their semantic similarity. The resulting graph is undirected if the semantic similarity measure between words is symmetric, which is the most common case. Such representation was already adopted for improving distributional thesauri by reranking the neighbors of their entries (Claveau et al., 2014) for instance.

One specificity of distributional thesauri from that perspective is that although the weight between two words is representative of their semantic similarity, we know from work such as (Ferret, 2010; Claveau et al., 2014) that the relevance of the semantic neighbors based on this weight strongly decreases as the rank of the neighbors increases. Consequently, our strategy for embedding distributional thesauri is two-fold: first, we build an embedding by relying on methods for embedding graphs, either by exploiting directly their structure or from their representation as matrices; second, we adapt the embedding resulting from the first step according to the specificities of distributional thesauri. We detail these two steps in the next two sections.

2.1 Graph Embedding

The problem of embedding graphs in the perspective of dimension reduction is not new and was already tackled by much work (Yan et al., 2007), going from spectral methods (Belkin and Niyogi, 2001) to more recently neural methods (Perozzi et al., 2014b; Cao et al., 2016). As graphs can be represented by their adjacency matrix, this problem is also strongly linked to the matrix factorization problem. The basic strategy is to perform the eigendecomposition of the matrix as for instance in the case of Latent Semantic Analysis (LSA) (Landauer and Dumais, 1997). However, such decomposition is computationally expensive and for large matrices, as in the context of Collaborative Filtering (Koren, 2008), less constrained matrix factorization techniques are used.

For turning a distributional thesaurus into word embeddings, we tested three different methods:

- the LINE algorithm (Tang et al., 2015), a recent method for embedding weighted graphs;
- the application of Singular Value Decomposition (SVD) to the adjacency matrix of the thesaurus;
- the matrix factorization approach proposed by Hu et al. (2008), also applied to the adjacency matrix of the thesaurus.

LINE defines a probabilistic model over the space $V \times V$, with V, the set of vertices of the considered graph. This probabilistic model is based on the representation of each vertex as a low-dimensional vector. This vector results from the minimization of an objective function based on the Kullback-Leibler divergence between the probabilistic model and the empirical distribution of the considered graph. This minimization is performed by the Stochastic Gradient Descent (SGD) method. Tang et al. (2015) propose more precisely two probabilistic models: one is based on the direct relation between two vertices while the second defines the proximity of two vertices according to the number of neighbors they share. We adopted the second model, which globally gives better results on several benchmarks.

In our second option, SVD factorizes T, the adjacency matrix of the thesaurus to embed, into the product $U \cdot \Sigma \cdot V^{\intercal}$. U and V are orthonormal and Σ is a diagonal matrix of eigenvalues. We classically adopted the truncated version of SVD by keeping only the first d elements of Σ, which finally leads to $T_d = U_d \cdot \Sigma_d \cdot V_d^{\intercal}$. Levy et al. (2015) investigated in the context of word co-occurrence matrices the best option for the low-dimensional representation of words as the usual setting was $U_d \cdot \Sigma_d$ while Caron (2001) suggested that $U_d \cdot \Sigma_d^P$ with $P < 1$ would be a better option. They found that $P = 0$ or $P = 0.5$ are clearly better than $P = 1$, with a slight superiority for $P = 0$. Similarly, we found $P = 0$ to be the best option.

Our last choice is based on a less constrained form of matrix factorization where T is decomposed into two matrices in such a way that $U \cdot V = \hat{T} \approx T$, with $T \in \mathbb{R}^{m \cdot n}$, $U \in \mathbb{R}^{m \cdot d}$, $V \in \mathbb{R}^{d \cdot n}$ and $d \ll m, n$. U and V are obtained by minimizing the following expression:

$$\sum_{i,j} (t_{ij} - u_i^\intercal v_j)^2 + \lambda(\|u_i\|^2 + \|v_j\|^2) \quad (1)$$

where the first term minimizes the reconstruction error of T by the product $U \cdot V$ while the second term is a regularization term, controlled by the parameter λ for avoiding overfitting. We used U as embedding of the initial thesaurus. (Hu et al., 2008) is a slight variation of this approach where t_{ij} is turned into a confidence score and the minimization of equation 1 is performed by the Alternating Least Squares method. One of the interests of this matrix factorization approach is its ability to deal with undefined values, which implements an implicit feedback in the context of recommender systems and can deal in our context with the fact that the input graph is generally sparse and does not include the furthest semantic neighbors of an entry.

2.2 From Graph to Thesaurus Embeddings

As mentioned previously, all the graph embedding methods of the previous section exploit the semantic similarity between words but for an entry, this similarity is not linearly correlated with the rank of its relevant neighbors in the thesaurus. In other words, the relevance of the semantic neighbors of an entry strongly decreases as their rank increases and the first neighbors are particularly important.

For taking into account this observation, we have adopted a strategy consisting in using the first neighbors of each entry of the initial thesaurus as constraints for adapting the embeddings built from this thesaurus by the graph embedding methods we consider. Such adaptation has already been tackled by some work in the context of the injection of external knowledge made of semantic relations into embeddings built mainly by neural methods such as the Skip-Gram model (Mikolov et al., 2013). Methods for performing such injection can roughly be divided into two categories: those operating during the building of the embeddings, generally by modifying the objective function supporting this building (Yih et al., 2012; Zhang et al., 2014), and those applied after the building of the embeddings (Yu and Dredze, 2014; Xu et al., 2014). We have more particularly used or adapted two methods from the second category and transposed one method from the first category for implementing our endogenous strategy.

The first method we have considered is the *retrofitting* method from Faruqui et al. (2015). This method performs the adaptation of a set of word vectors q_i by minimizing the following objective function through a label propagation algorithm (Bengio et al., 2006):

$$\sum_{i=1}^{n} \left[\|q_i - \hat{q}_i\|^2 + \sum_{(i,j) \in E} \|q_i - q_j\|^2 \right] \quad (2)$$

where $\hat{q}_i$ are the q_i vectors after their adaptation. The first term is a stability term ensuring that the adapted vectors do not diverge too much from the initial vectors while the second term represents an adaptation term, tending to bring closer the vectors associated with words that are part of a relation from an external knowledge source E. In our case, this knowledge corresponds to the relations between each entry of the initial thesaurus and its first neighbors.

The second method, *counter-fitting* (Mrkšić et al., 2016), is close to *retrofitting* and mainly differentiates from it by adding to the objective function a repelling term for pushing vectors corresponding to antonymous words away from each other. However, a distributional thesaurus does not contain identified antonymous words[1]. Hence, we discarded this term and used the following objective function, minimized by SGD:

$$\begin{aligned} &\sum_{i=1}^{N} \sum_{j \in N(i)} \tau(dist(\hat{q}_i, \hat{q}_j) - dist(q_i, q_j)) \\ &+ \sum_{(i,j) \in E} \tau(dist(\hat{q}_i, \hat{q}_j)) \end{aligned} \quad (3)$$

with $dist(x, y) = 1 - \cos(x, y)$ and $\tau(x) = \max(0, x)$. As in equation 2, the first term tends to preserve the initial vectors. In this case, this preservation does not focus on the vectors themselves but on the pairwise distances between a vector and its nearest neighbors ($N(i)$). The second term is quite similar to the second term of equation 2 with the use of a distance derived from the Cosine similarity instead of the Euclidean distance[2].

[1] We tried to exploit semantic neighbors that are not very close to their entry as antonyms but results were globally better without them.

[2] Since the Cosine similarity is used as similarity measure between words through their vectors, this distance should be more adapted in this context than the Euclidean distance.

The last method we have used for improving the embeddings built from the initial thesaurus, called *rank-fitting* hereafter, is a transposition of the method proposed by Liu et al. (2015). The objective of this method is to integrate into embeddings order constraints coming from external knowledge with the following form: similarity$(w_i, w_j) >$ similarity(w_i, w_k), abbreviated $s_{ij} > s_{ik}$ in what follows. This kind of constraints particularly fits our context as the semantic neighbors of an entry in a distributional thesaurus are ranked and can be viewed as a set of such constraints. More precisely, i corresponds in this case to an entry and j and k to two of its neighbors such that *rank(j) > rank(k)*. However, the method of Liu et al. (2015) is linked to the Skip-Gram model and was defined as a modification of the objective function underlying this model. We have transposed this approach for its application to the adaptation of embeddings after their building, without a specific link to the Skip-Gram model.

The general idea is to adapt vectors to minimize $s_{ij} - s_{ik} \quad \forall (i, j, k) \in E$. The objective to minimize takes more specifically the following form:

$$\sum_{(i,j,k)\in E} f(s_{ik} - s_{ij}) \tag{4}$$

where $f(s_{ik} - s_{ij}) = \max(0, s_{ik} - s_{ij})$ corresponds to a kind of hinge loss function and the similarity between words i and j, s_{ij}, is given by the Cosine measure between their associated vectors. The minimization of this objective is performed as for *counter-fitting* by SGD.

Finally, we have also defined a mixed *counter-rank-fitting* method that associates constraints about the proximity of word vectors and their relative ranking. This association was done by mixing the objective functions of *counter-fitting* and *rank-fitting* through the addition of the second term of equation 3, *i.e.* its adaptation term, and equation 4. In this configuration, the first term of the *counter-fitting* function, that preserves the initial embeddings, was not found useful anymore in preliminary experiments.

3 Evaluation of Thesaurus Embedding

3.1 Experimental Framework

For testing and evaluating the proposed approach, we needed first to choose a reference corpus and to build a distributional thesaurus from it. We chose the AQUAINT-2 corpus, already used for various evaluations, a middle-size corpus of around 380 million words made of news articles in English. The main preprocessing of the corpus was the application of lemmatization and the removal of function words. According to (Bullinaria and Levy, 2012), the lemmatization of words leads to only a small improvement in terms of results but it is also a way to obtain the same results with a smaller corpus.

The building of our reference distributional thesaurus, T$_{cnt}$, was achieved by relying on a classical count-based approach with a set of parameters that were found relevant by several systematic studies (Baroni et al., 2014; Kiela and Clark, 2014; Levy et al., 2015):

- distributional contexts: co-occurrents restricted to nouns, verbs and adjectives having at least 10 occurrences in the corpus, collected in a 3 word window, *i.e.* +/-1 word around the target word;
- directional co-occurrents, which were found having a good performance by Bullinaria and Levy (2012);
- weighting function of co-occurrents in contexts = *Positive Pointwise Mutual Information* (PPMI) with the *context distribution smoothing* factor proposed by (Levy et al., 2015), equal to 0.75;
- similarity measure between contexts, for evaluating the semantic similarity of two words = *Cosine* measure;
- filtering of contexts: removal of co-occurrents with only one occurrence.

The building of the thesaurus from the distributional data was performed as in (Lin, 1998) or (Curran and Moens, 2002) by extracting the closest semantic neighbors of each of its entries. More precisely, the similarity measure was computed between each entry and its possible neighbors. Both the entries of the thesaurus and their possible neighbors were nouns with at least 10 occurrences in the corpus. These neighbors were then ranked in the decreasing order of the values of this measure.

The evaluation of distributional objects such as thesauri or word embeddings is currently a subject of research as both intrinsic (Faruqui et al., 2016; Batchkarov et al., 2016) and extrinsic (Schnabel et al., 2015) evaluations exhibit insufficiencies that question their reliability. In our case, we per-

Method	#eval. words	#syn./ word	R@100	R_{prec}	MAP	P@1	P@2	P@5
T_{cnt}	10,544	2.9	29.0	11.3	13.1	15.7	11.4	6.6
GloVe			21.3	6.7	8.0	9.8	7.4	4.5
SGNS			22.4	8.7	10.3	12.3	8.8	5.2

Table 1: Evaluation of the initial thesaurus and two reference models of embeddings (values x 100)

formed an intrinsic evaluation relying on the synonyms of WordNet 3.0 (Miller, 1990) as Gold Standard. This choice was first justified by our overall long-term perspective, illustrated in Section 5, which is the extraction of synonyms from documents and the expansion of already existing sets of synonyms. However, it is also likely to alleviate some evaluation problems as it narrows the scope of the evaluation, by restricting to a specific type of semantic relations, but performs it at a large scale, the combination of which making its results more reliable. For focusing on the evaluation of the extracted semantic neighbors, the WordNet 3.0's synonyms were filtered to discard entries and synonyms that were not part of the AQUAINT-2 vocabulary. The number of evaluated words and the average number of synonyms in our Gold Standard for each entry are given by the second and the third columns of Table 1.

In terms of methodology, the kind of evaluation we have performed follows (Curran and Moens, 2002; Ferret, 2010) by adopting an Information Retrieval point of view in which each entry is considered as a query and its neighbors are viewed as retrieved synonyms. Hence, we adopted the classical evaluation measures in the field: the R-precision (R_{prec}) is the precision after the first R neighbors were retrieved, R being the number of Gold Standard synonyms; the Mean Average Precision (MAP) is the mean of the precision values each time a Gold Standard synonym is found; precision at different cut-offs is given for the 1, 2, 5 first neighbors. We also give the global recall for the first 100 neighbors.

Table 1 shows the evaluation according to these measures of our initial distributional thesaurus T_{cnt} along with the evaluation in the same framework of two reference models for building word embeddings from texts: GloVe from Pennington et al. (2014) and Skip-Gram with negative sampling (SGNS) from Mikolov et al. (2013)[3]. The input of these two models was the lemmatized version of the AQUAINT-2 corpus as for T_{cnt} but with all its words. Each model was built with the best parameters found from previous work and tested on this corpus. For GloVe: vectors of 300 dimensions, window size = 10, addition of word and context vectors and 100 iterations; for SGNS: vectors of 400 dimensions, window size = 5, 10 negative examples and default value for down-sampling of highly frequent words.

Two main trends can be drawn from this evaluation. First, T_{cnt} significantly outperforms GloVe and SGNS for all measures[4]. This superiority of a count-based approach over two predict-based approaches can be seen as contradictory with the findings of Levy et al. (2015). Our analysis is that the use of directional co-occurrences, a rarely tested parameter, explains a large part of this superiority. The second conclusion is that SGNS significantly outperforms GloVe for all measures. Hence, we will report results hereafter only for SGNS as a reference word embedding model.

3.2 Graph Embedding Evaluation

We have evaluated the three methods presented in Section 2.1 for embedding our initial thesaurus T_{cnt} according to the evaluation framework presented in the previous section. For all methods, the main parameters were the number of neighbors taken into account and the number of dimensions of the final vectors. In all cases, the number of neighbors was equal to 5,000, LINE being not very affected by this parameter, and the size of the vectors was 600[5]. For LINE, 10 billion samplings of the similarity values were done and for the matrix factorization (MF) approach, we used $\lambda = 0.075$.

According to Table 2, SVD significantly appears as the best method even if LINE is a competitive alternative. SVD outperforms GloVe while

[3] Following (Levy et al., 2015), SGNS was preferred to the Continuous Bag-Of-Word (CBOW) model.

[4] The statistical significance of differences were judged according to a paired Wilcoxon test with p-value < 0.05. The same test was applied for results reported hereafter.

[5] The values of these parameters were optimized on another thesaurus, coming from (Ferret, 2010).

Method	**R$_{prec}$**	**MAP**	**P@1**	**P@2**	**P@5**
T$_{cnt}$	11.3	13.1	15.7	11.4	6.6
SGNS	8.7	10.3	12.3	8.8	5.2
SVD	**7.8**	**9.5**	**11.3**	**8.1**	**5.0**
LINE	6.8	8.3	9.7	7.1	4.4
MF	4.0	4.9	5.9	4.4	2.7

Table 2: Evaluation of the embedding of a thesaurus as a graph

Method	**R$_{prec}$**	**MAP**	**P@1**	**P@2**	**P@5**
T$_{cnt}$	11.3	13.1	15.7	11.4	6.6
SGNS	8.7	10.3	12.3	8.8	5.2
SVD	7.8	9.5	11.3	8.1	5.0
Retrofit	**10.9**	**12.9**	**15.2**	11.4	6.8
Counterfit	10.6	12.8	14.0	**11.9**	**7.3**
Rankfit	9.0	10.5	12.6	9.0	5.3
Counter-rankfit	10.7	12.4	**15.2**	11.0	6.3

Table 3: Evaluation of the global thesaurus embedding process

LINE is equivalent to it, which is a first interesting result: this first embedding step of a distributional thesaurus is already able to produce better word representations than a state-of-the-art method, even if it does not reach the level of the best one (SGNS). However, Table 2 also shows that there is still room for improvement for reaching the level of the initial thesaurus T$_{cnt}$. Finally, the matrix factorization approach is obviously a bad option, at least under the tested form.

3.3 Thesaurus Embedding Evaluation

Table 3 shows the results of the evaluation of the word embedding adaptation methods of Section 2.2, which is also the evaluation of the global thesaurus embedding process. For all methods, the input embeddings were produced by applying SVD to the initial thesaurus T$_{cnt}$, which was shown as the best option by Table 2. For *retrofitting* (Retrofit) and *counter-fitting* (Counterfit), only the relations between each entry of the thesaurus and its first and second neighbors were considered. For *rank-fitting* (Rankfit), the neighborhood was extended to the first 50 neighbors. For the optimization processes, we used the default settings of the methods: 10 iterations for *retrofitting* and 20 iterations for *counter-fitting*. We also used 20 iterations for *rank-fitting* and *counter-rank-fitting* (Counter-rankfit). For all optimizations by SGD, the learning rate was 0.01.

Several observations can be done. First, all the tested methods significantly improve the initial embeddings. Second, the results of the different methods are quite close for all measures. *retrofitting* outperforms *counter-fitting* but not significantly for R$_{prec}$. *rank-fitting* is significantly the worst method and its association with *counter-fitting* is better than *retrofitting* for P@1 only, but not significantly. However, we can globally note that the association of SVD and the best adaptation methods obtains results close to the results of the initial T$_{cnt}$ (the difference is even not significant for R$_{prec}$ and P@5). As a consequence, we can conclude, in connection with our initial objective, that embedding a distributional thesaurus while preserving its information in terms of semantic similarity is possible.

4 Applications of Thesaurus Embedding

4.1 Improvement of Existing Embeddings

In the previous section, we have shown that the strongest relations of a distributional thesaurus can be used for improving word vectors built from the embedding of this thesaurus. Since this adaptation is performed after the building of the vectors, it can actually be applied to all kinds of embeddings elaborated from the corpus used for building the distributional thesaurus. As for the process of the previous section, this is a kind of bootstrapping approach in which the knowledge extracted from a corpus is used for improving the word representations elaborated from this corpus. Moreover, as GloVe and most word embedding models, SGNS relies on first-order co-occurrences between words. From that perspective, adapting SGNS embeddings with relations coming from a distributional thesaurus built from the same corpus as these embeddings is a way to incorporate second-order co-occurrence relations into them.

Method	**R$_{prec}$**	**MAP**	**P@1**	**P@2**	**P@5**
(S)GNS	8.7	10.3	12.3	8.8	5.2
Emb$_{retrof}$(T$_{cnt}$)	10.9	12.9	15.2	11.4	6.8
S+Counter-rankfit	**9.5**	**11.1**	**13.8**	**9.9**	**5.6**
S+Retrofit	9.3	10.6	13.2	9.6	5.5

Table 4: Evaluation of the adaptation of SGNS embeddings with thesaurus relations

For this experiment, we applied both *retrofitting* and *counter-rank-fitting* with exactly the same parameters as in Section 3.3. The results of Table 4 clearly validate the benefit of the technique: both *retrofitting* and *counter-rank-fitting* significantly improve SGNS embeddings. As in Section 3.3, the results of *retrofitting* and *counter-rank-fitting* are rather close, with a global advantage for *counter-rank-fitting*. We can also note that the improved versions of SGNS embeddings are still far from the best results of our thesaurus embedding method (*SVD* + *Retrofit*).

4.2 Fusion of Heterogeneous Representations

Being able to turn a distributional thesaurus into word embeddings also makes it possible to fusion different types of distributional data. In the case of thesaurus, fusion processes were early proposed by Curran (2002) and more recently by Ferret (2015). In the case of word embeddings, the recent work of Yin and Schütze (2016) applied ensemble methods to several word embeddings. By exploiting the possibility to change from one type of representation to another, we propose a new kind of fusion, performed between a thesaurus and word embeddings and leading to improve both the input thesaurus and the embeddings.

The first step of this fusion process consists in turning the input word embeddings into a distributional thesaurus. Then, the resulting thesaurus is merged with the input thesaurus, which consists in merging two lists of ranked neighbors for each of their entries. We followed (Ferret, 2015) and applied for this fusion the CombSum strategy to the similarity values between entries and their neighbors, normalized with the Zero-one method (Wu et al., 2006). Finally, we applied the method of Section 2 for turning the thesaurus resulting from this fusion into word embeddings.

Method	R$_{prec}$	MAP	P@1	P@2	P@5
(T)$_{cnt}$	11.3	13.1	15.7	11.4	6.6
(S)GNS	8.7	10.3	12.3	8.8	5.2
Fusion T-S	12.5	14.8	17.2	12.8	7.5
Emb$_{retrof}$(fusion T-S)	11.8	13.8	16.7	12.4	7.0

Table 5: Evaluation of the fusion of a distributional thesaurus T and word embeddings S

The evaluation of this fusion process, performed in a shared context as the considered thesaurus and word embeddings are built from the same corpus, is given in Table 5. The *Fusion T-S* line corresponds to the evaluation of the thesaurus resulting from the second step of the fusion process. The significant difference with the results of T$_{cnt}$ and SGNS confirms the conclusions of Ferret (2015) about the interest of merging thesauri built differently. The *Emb$_{retrof}$(fusion T-S)* line shows the evaluation of the word embeddings produced by the global fusion process. In a similar way to the findings of Section 3.3, the embeddings built from the *Fusion T-S* thesaurus are less effective than the thesaurus itself but the difference is small here too. Moreover, we can note that these embeddings have significantly higher results than SGNS, the input embeddings, but also higher results than the input thesaurus T$_{cnt}$, once again without any external knowledge.

5 Knowledge Injection and Synset Expansion

In this section, we will illustrate how the improvement of a distributional thesaurus, obtained in our case by the injection of external knowledge, can be transposed to word embeddings. Moreover, we will show that the thesaurus embedding process achieving this transposition obtains better results for taking into account external knowledge than methods, such as *retrofitting*, that are applied to embeddings built directly from texts (SGNS in our case). We will demonstrate this superiority more precisely in the context of synset expansion.

The overall principle is quite straightforward: first, the external knowledge is integrated into a distributional thesaurus built from the source corpus (T$_{cnt}$ in our experiments). Then, the resulting thesaurus is embedded following the method of Section 2. This external knowledge is supposed to be made of semantic similarity relations. We have considered more particularly pairs of synonyms (E, K) such that E is an entry of T$_{cnt}$ and K is a synonym of E randomly selected from the WordNet 3.0's synsets E is part of. Each E is part of only one pair (E, K).

5.1 Injecting External Knowledge into a Thesaurus

The integration of the semantic relations into a distributional thesaurus is done for each entry E by reranking the neighbor K of the (E, K) pair at the highest rank with the highest similarity. The line T_{cnt}+K of Table 6 gives the evaluation of this integration for 10,544 pairs (E, K) of synonyms, which means one synonym by entry.

	Evaluation of memorization					Global evaluation				
Method	**R$_{prec}$**	**MAP**	**P@1**	**P@2**	**P@5**	**R$_{prec}$**	**MAP**	**P@1**	**P@2**	**P@5**
SGNS	6.5	9.7	6.5	4.6	2.6	8.7	10.3	12.3	8.8	5.2
SGNS+retrof(K)	82.4	90.3	82.4	47.8	19.9	80.1	82.0	98.1	72.3	36.9
T$_{cnt}$	8.5	12.4	8.5	5.9	3.2	11.3	13.1	15.7	11.4	6.6
svd(T$_{cnt}$)	5.8	9.0	5.8	4.0	2.3	7.8	9.5	11.3	8.1	5.0
svd(T$_{cnt}$)+retrof(K)	86.6	92.8	86.6	48.8	20.0	81.5	83.5	98.8	72.6	37.4
T$_{cnt}$+K	100	100	100	50.0	20.0	62.7	63.8	100	54.0	23.1
svd(T$_{cnt}$+K)	12.0	18.0	12.0	8.3	4.7	13.8	17.1	19.0	13.7	8.1
svd(T$_{cnt}$+K)+retrof(K)	**88.3**	**93.9**	**88.3**	**49.2**	**20.0**	**82.6**	**84.5**	**99.5**	**73.2**	**38.2**

Table 6: Evaluation of the injection of external knowledge into word embeddings for synset expansion

As our evaluation methodology is based on the synonyms of WordNet, we have split our evaluation in two parts. One part takes as Gold Standard the synonyms used for the knowledge injection (see the *Evaluation of memorization* columns in Table 6) and evaluates to what extent the injected knowledge has been memorized. The second part (see the *Global evaluation* columns in Table 6) considers all the synonyms used for the evaluations in the previous sections as Gold Standard for evaluating the ability of models not only to memorize the injected knowledge but also to retrieve new synonyms, *i.e.* synonyms that are not part of the injected knowledge. In the context of our evaluation, which is based on synonym retrieval, this kind of generalization can also be viewed as a form of synset expansion. This is another way to extract synonyms from texts compared to work such as (Leeuwenberg et al., 2016; Minkov and Cohen, 2014; van der Plas and Tiedemann, 2006).

In the case of *T_{cnt}+K*, we can note that the memorization is perfect, which is not a surprise since the injection of knowledge into the thesaurus corresponds to a kind of memorization. No specific generalization effect beyond the synonyms already present in the thesaurus is observed for the same reason.

5.2 From a Knowledge-Boosted Thesaurus to Word Embeddings

The result of the process described in the previous section is what we could call a knowledge-boosted distributional thesaurus. However, its form is not different from a classical distributional thesaurus and it can be embedded similarly by applying the method of Section 2. The only difference with this method concerns its second step: instead of leveraging the first n neighbors of each entry for improving the embeddings obtained by SVD, we exploited the set of relations used for "boosting" the initial thesaurus.

The evaluation of the new method we propose for building word embeddings integrating external knowledge is presented in Table 6. More precisely, three different methods are compared: a state-of-the-art method, *SGNS+retrof(K)*, consisting in applying *retrofitting* to SGNS embeddings. *retrofitting* was chosen as it is quick and gives good results. The second method, *svd(T_{cnt})+retrof(K)*, applies *retrofitting* to the embeddings built from T$_{cnt}$ by SVD. The last method, *svd(T_{cnt}+K)+retrof(K)*, corresponds to the full process we have presented, where the external knowledge is first injected into the initial thesaurus T$_{cnt}$ before its embedding.

First, we can note that all the methods considered for producing word embeddings by taking into account external knowledge leads to a very strong improvement of results compared to their starting point. This is true both for the memorization and global evaluations. From the memorization viewpoint, all the injected synonyms can be found among the first five neighbors returned by the three methods as illustrated by their P@5 and even at the first rank in nearly nine times out of ten for the best method, which is clearly our thesaurus embedding process (except the pure memorization performed by T$_{cnt}$+K).

We can also observe that the method used for knowledge injection can reverse initial differences. For instance, the application of SVD to a thesaurus built from a corpus, *svd(T_{cnt})*, obtains lower results than the application of SGNS to the same corpus. After the injection of external knowledge, this ranking is reversed: the values of the evaluation measures are higher for *svd(T_{cnt})+retrof(K)* than for *SGNS+retrofit(K)*.

More importantly, Table 6 shows that the inte-

Entries	**K**	**Synonyms in neighbors of T$_{cnt}$+K**	**Synonyms in neighbors of svd(T$_{cnt}$+K)+retrof(K)**
richness	fullness	fullness [1] 1.0, affluence [1,665] 0.06, profusion [1,950] 0.06, fertility [2,000] 0.06, cornucopia [2,919] 0.06	fullness [1] 0.80, affluence [2] 0.71, cornucopia [3] 0.71, fertility [5] 0.66, profusion [6] 0.44
butchery	abattoir	abattoir [1] 1.0, slaughterhouse [2] 0.05, carnage [65] 0.03, slaughter [90] 0.03, massacre [132] 0.03, shambles [3,735] 0.02	abattoir [1] 0.64, massacre [2] 0.62, carnage [3] 0.61, slaughterhouse [4] 0.53, shambles [5] 0.45, slaughter [11] 0.21
idiom	dialect	dialect [1] 1.0, phrase [16] 0.09, accent [62] 0.09, parlance [2,971] 0.07	dialect [1] 0.80, phrase [2] 0.75, accent [3] 0.71, parlance [4] 0.71
spectator	witness	witness [1] 1.0, viewer [28] 0.14, watcher [519] 0.12	watcher [1] 0.59, witness [2] 0.56, viewer [3] 0.51, looker [10] 0.30

Table 7: Examples of the interest of thesaurus embedding for synset expansion. Each synonym is given with its [rank] among the neighbors of the entry and its similarity value with the entry

gration of external knowledge into the thesaurus before its embedding is clearly effective as illustrated by the significant differences between *SGNS+retrofit(K)* and *svd(T_{cnt}+K)+retrof(K)*. Finally, from the synset expansion viewpoint, it is worth adding that the P@2 value of our best method means that the first synonym proposed by the expansion in addition to the injected synonyms is correct with a precision equal to 46.9, which represents 4,945 new synonyms and illustrates the generalization capabilities of the method.

Table 7 illustrates more qualitatively for some words the interest of the thesaurus embedding method we propose for the expansion of existing synsets. In accordance with the findings of Table 6, it first shows that the method has a good memorization capability of the injected knowledge (K) in the initial thesaurus since in the resulting embeddings (*svd(T_{cnt}+K)+retrof(K)*), the synonym provided for each entry appears as the first or the second neighbor.

Table 7 also illustrates the good capabilities of the method observed in Table 6 in terms of generalization as the rank of synonyms of an entry not provided as initial knowledge tend to decrease strongly. For instance, for the entry *idiom*, the rank of the synonym *parlance* is equal to 2,971 in the initial thesaurus with the injected knowledge (T_{cnt}+K) while it is only equal to 4 after the embedding of the thesaurus. Interestingly, this improvement in terms of rank comes from a change in the distributional representation of words that also impacts the evaluation of the semantic similarity between words. While the similarity between the word *richness* and its synonym *profusion* was initially very low (0.06), its value after the embedding process is very much higher (0.66) and more representative of the relation between the two words.

6 Conclusion and Perspectives

In this article, we presented a method for building word embeddings from distributional thesauri with a limited loss of semantic similarity information. The resulting embeddings outperforms state-of-the-art embeddings built from the same corpus. We also showed that this method can improve already existing word representations and the injection of external knowledge into word embeddings.

A first extension to this work would be to better leverage the ranking of neighbors in a thesaurus and to integrate more tightly the two steps of our embedding method. We also would like to define a more elaborated method for injecting external knowledge into a distributional thesaurus, more precisely by exploiting the injected knowledge to rerank its semantic neighbors. Finally, we would be interested in testing further the capabilities of the embeddings with injected knowledge for extending resources such as WordNet.

References

Marco Baroni, Georgiana Dinu, and Germán Kruszewski. 2014. Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors. In *52^{nd} Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 238–247, Baltimore, Maryland.

Miroslav Batchkarov, Thomas Kober, Jeremy Reffin, Julie Weeds, and David Weir. 2016. A critique of word similarity as a method for evaluating distributional semantic models. In *1st Workshop on Evalu-*

ating Vector-Space Representations for NLP, pages 7–12, Berlin, Germany.

Mikhail Belkin and Partha Niyogi. 2001. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In *Advances in Neural Information Processing Systems 14*, pages 585–591.

Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. 2006. Label Propagation And Quadratic Criterion. In *Semi-Supervised Learning*, pages 193–216. MIT Press.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based Measures of Lexical Semantic Relatedness. *Computational Linguistics*, 32(1):13–47.

John A Bullinaria and Joseph P Levy. 2012. Extracting semantic representations from word co-occurrence statistics: stop-lists, stemming, and SVD. *Behavior research methods*, 44(3):890–907.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2016. Deep Neural Networks for Learning Graph Representations. In *Thirtieth AAAI Conference on Artificial Intelligence (AAAI 2016)*, pages 1145–1152. AAAI Press.

John Caron. 2001. Computational Information Retrieval. chapter Experiments with LSA Scoring: Optimal Rank and Basis, pages 157–169. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA.

Vincent Claveau, Ewa Kijak, and Olivier Ferret. 2014. Improving distributional thesauri by exploring the graph of neighbors. In *25^{th} International Conference on Computational Linguistics (COLING 2014)*, pages 709–720, Dublin, Ireland.

James Curran. 2002. Ensemble Methods for Automatic Thesaurus Extraction. In *2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 222–229.

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

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting Word Vectors to Semantic Lexicons. In *2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2015)*, pages 1606–1615, Denver, Colorado.

Manaal Faruqui, Yulia Tsvetkov, Pushpendre Rastogi, and Chris Dyer. 2016. Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany.

Olivier Ferret. 2010. Testing Semantic Similarity Measures for Extracting Synonyms from a Corpus. In *7^{th} International Conference on Language Resources and Evaluation (LREC'10)*, pages 3338–3343, Valletta, Malta.

Olivier Ferret. 2015. Early and Late Combinations of Criteria for Reranking Distributional Thesauri. In *53^{rd} Annual Meeting of the Association for Computational Linguistics and 7^{th} International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 470–476, Beijing, China.

Gregory Grefenstette. 1994. *Explorations in automatic thesaurus discovery*. Kluwer Academic Publishers.

Y. Hu, Y. Koren, and C. Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining (ICDM'08)*, pages 263–272.

Douwe Kiela and Stephen Clark. 2014. A Systematic Study of Semantic Vector Space Model Parameters. In *2^{nd} Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*, pages 21–30, Gothenburg, Sweden.

Yehuda Koren. 2008. Factorization Meets the Neighborhood: A Multifaceted Collaborative Filtering Model. In *14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2008)*, pages 426–434.

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

Artuur Leeuwenberg, Mihaela Vela, Jon Dehdari, and Josef van Genabith. 2016. A Minimally Supervised Approach for Synonym Extraction with Word Embeddings. *The Prague Bulletin of Mathematical Linguistics*, (105):111–142.

Omer Levy, Yoav Goldberg, and Ido Dagan. 2015. Improving Distributional Similarity with Lessons Learned from Word Embeddings. *Transactions of the Association for Computational Linguistics (TALC)*, 3:211–225.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *17^{th} International Conference on Computational Linguistics and 36^{th} Annual Meeting of the Association for Computational Linguistics (ACL-COLING'98)*, pages 768–774, Montral, Canada.

Quan Liu, Hui Jiang, Si Wei, Zhen-Hua Ling, and Yu Hu. 2015. Learning Semantic Word Embeddings based on Ordinal Knowledge Constraints. In *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2015)*, pages 1501–1511, Beijing, China.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations 2013 (ICLR 2013), workshop track*.

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

Einat Minkov and William W. Cohen. 2014. Adaptive graph walk-based similarity measures for parsed text. *Natural Language Engineering*, 20(3):361397.

Nikola Mrkšić, Diarmuid Ó Séaghdha, Blaise Thomson, Milica Gašić, Lina M. Rojas-Barahona, Pei-Hao Su, David Vandyke, Tsung-Hsien Wen, and Steve Young. 2016. Counter-fitting Word Vectors to Linguistic Constraints. In *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016)*, pages 142–148, San Diego, California.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1532–1543, Doha, Qatar.

Bryan Perozzi, Rami Al-Rfou, Vivek Kulkarni, and Steven Skiena. 2014a. *Inducing Language Networks from Continuous Space Word Representations*. Springer International Publishing, Bologna, Italy.

Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014b. DeepWalk: Online Learning of Social Representations. In *20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2014)*, pages 701–710.

Lonneke van der Plas and Jörg Tiedemann. 2006. Finding Synonyms Using Automatic Word Alignment and Measures of Distributional Similarity. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 866–873, Sydney, Australia.

Tobias Schnabel, Igor Labutov, David Mimno, and Thorsten Joachims. 2015. Evaluation methods for unsupervised word embeddings. In *2015 Conference on Empirical Methods in Natural Language Processing (EMNLP 2015)*, pages 298–307, Lisbon, Portugal.

Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. LINE: Large-scale Information Network Embedding. In *24th International Conference on World Wide Web (WWW 2015)*, WWW '15, pages 1067–1077.

Shengli Wu, Fabio Crestani, and Yaxin Bi. 2006. Evaluating Score Normalization Methods in Data Fusion. In *Third Asia Conference on Information Retrieval Technology (AIRS'06)*, pages 642–648. Springer-Verlag.

Chang Xu, Yalong Bai, Jiang Bian, Bin Gao, Gang Wang, Xiaoguang Liu, and Tie-Yan Liu. 2014. RC-NET: A General Framework for Incorporating Knowledge into Word Representations. In *23rd ACM International Conference on Conference on Information and Knowledge Management (CIKM 2014)*, pages 1219–1228.

S. Yan, D. Xu, B. Zhang, H. j. Zhang, Q. Yang, and S. Lin. 2007. Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(1):40–51.

Wen-tau Yih, Geoffrey Zweig, and John Platt. 2012. Polarity Inducing Latent Semantic Analysis. In *2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2012)*, pages 1212–1222, Jeju Island, Korea.

Wenpeng Yin and Hinrich Schütze. 2016. Learning Word Meta-Embeddings. In *54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1351–1360, Berlin, Germany.

Mo Yu and Mark Dredze. 2014. Improving Lexical Embeddings with Semantic Knowledge. In *52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014)*, pages 545–550, Baltimore, Maryland.

Jingwei Zhang, Jeremy Salwen, Michael Glass, and Alfio Gliozzo. 2014. Word Semantic Representations using Bayesian Probabilistic Tensor Factorization. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP 2014)*, pages 1522–1531, Doha, Qatar.