Improving distributional thesauri by exploring the graph of neighbors

Vincent Claveau, Ewa Kijak, Olivier Ferret

Improving distributional thesauri by exploring the graph of neighbors

Vincent Claveau
IRISA - CNRS
Campus de Beaulieu
35042 Rennes, France
vincent.claveau@irisa.fr

Ewa Kijak
IRISA - Univ. of Rennes 1
Campus de Beaulieu
35042 Rennes, France
ewa.kijak@irisa.fr

Olivier Ferret
CEA, LIST
LVIC
91191 Gif-sur-Yvette, France
olivier.ferret@cea.fr

Abstract

In this paper, we address the issue of building and improving a distributional thesaurus. We first show that existing tools from the information retrieval domain can be directly used in order to build a thesaurus with state-of-the-art performance. Secondly, we focus more specifically on improving the obtained thesaurus, seen as a graph of k-nearest neighbors. By exploiting information about the neighborhood contained in this graph, we propose several contributions. 1) We show how the lists of neighbors can be globally improved by examining the reciprocity of the neighboring relation, that is, the fact that a word can be close of another and vice-versa. 2) We also propose a method to associate a confidence score to any lists of nearest neighbors (i.e. any entry of the thesaurus). 3) Last, we demonstrate how these confidence scores can be used to reorder the closest neighbors of a word. These different contributions are validated through experiments and offer significant improvement over the state-of-the-art.

1 Introduction

Distributional thesauri are useful for many NLP tasks and their construction is an issue widely discussed for several years (Grefenstette, 1994). However this is still a very active research field, maintained by the increasingly large number of available corpus and by many applications. These thesauri associate each of their entry with a list of words that are desired semantically close to the entry. This notion of proximity varies (synonymy, other paradigmatic relations, syntagmatic relations (Budanitsky and Hirst, 2006; Adam et al., 2013, for a discussion)), but the methods used for the automatic construction of thesauri are often shared. For the most part, these methods rely on the distributional hypothesis of (Firth, 1957): each word is characterized by the set of contexts in which it appears, and the semantic proximity of two words can be inferred from the proximity of their contexts. This hypothesis has been implemented in different ways, and several propositions to improve the results have been explored (see next section for a state of the art).

The work presented in this article are part of this framework. We propose several contributions on the creation of these distributional thesauri and their improvement. We first show that models from information retrieval (IR) can provide information on semantic relationships, and are thus adapted to the task of creating these thesauri. In addition, they offer very competitive results compared to the state of the art, while enjoying existing tools (Section 3).

The most important part of our work then focuses on the exploitation of such semantic neighborhood relations. The IR models indeed provide lists ordering all words by decreasing similarity, that form a graph of nearest neighbors. We propose to take advantage of some of the neighborhood information contained in this graph and we derive three contributions.
1) We globally improve neighbor lists by taking into account the reciprocity of the neighborhood relationship, that is to say the fact that a word is a close neighbor of another and vice versa (Section 4).
2) We also propose a method that associates each neighbor list (i.e. each entry of the thesaurus built) with a confidence score (Section 5). This method uses the nearest neighbor graph to estimate the probabilities that a given word is the i-th neighbor of another word.

3) Finally, on the basis of this work, we show how to use this confidence score and these probabilities to reorder the list of nearest neighbors (Section 6). To achieve this goal, we model the reranking as an optimization problem of assignments, solved by the Hungarian algorithm (Kuhn and Yaw, 1955).

2 Related work

The notion of distributional thesaurus, as it was initially defined by Grefenstette (1994), followed by Lin (1998a) and Curran and Moens (2002), is not often considered specifically, probably because of its strong link with the notion of semantic similarity. As a consequence, the improvement of distributional thesauri has been first a side effect of the improvement of the distributional similarity measures used for their building and more precisely, of the distributional data they rely on. Both the nature of the constituents of distributional contexts and their weighting have been considered in this regard. Concerning their weighting, Broda et al. (2009) proposed to turn the weights of context constituents into ranks to make them less dependent on a specific weighting function while Zhitomirsky-Geffet and Dagan (2009), extended by Yamamoto and Asakura (2010), defined a bootstrapping method for modifying the weights of constituents in the distributional context of a word according to the similarity with its semantic neighbors.

The nature of distributional contexts has been first considered through the distinction between window-based and syntactic co-occurrents (Grefenstette, 1994; Curran and Moens, 2002). However, most of the work related to this issue has focused on the fact that the "traditional" representation of distributional contexts is very sparse and redundant, as illustrated by Hagiwara et al. (2006). Hence, several methods for dimension reduction were tested in this context: from Latent Semantic Analysis (Landauer and Dumais, 1997), extended for syntactic co-occurrents (Padó and Lapata, 2007), to Random Indexing (Sahlgren, 2001), Non-negative Matrix Factorization (Van de Cruys, 2010) and more recently, lexical representations learnt by neural networks (Huang et al., 2012; Mikolov et al., 2013).

The work we present in this article follows a different perspective as our objective is to improve an existing distributional thesaurus by relying on its structure through a reranking of its neighbors. Such approach was adopted to some extent by Zhitomirsky-Geffet and Dagan (2009) as it exploited the neighbors of an entry in an initial thesaurus for reweighting its distributional representation and finally, reranking its neighbors. Ferret (2013) proposed a more indirect method in which the reranking is based on the downgrading of the neighbors that are detected as not similar to their entry through a pseudo word sense disambiguation task: such detection occurs if a certain proportion of the occurrences of a neighbor are not tagged as the entry. Finally, the closest work to ours is (Ferret, 2012), which selects in an unsupervised way a set of examples of semantically similar words from an initial thesaurus for training a classifier whose decision function is used for reranking the neighbors of each entry. Its unsupervised selection of examples is more precisely based on the symmetry of semantic similarity relations.

As Ferret (2012), our work exploits a certain kind of symmetry in the relation of distributional neighborhood between words but extends it to a larger scale by considering the initial thesaurus as a k-nearest neighbor graph and using the relations in this graph for reranking the neighbors of each entry, similarly to Pedronette et al. (2014) in the context of image retrieval.

3 IR models for building distributional thesauri

3.1 Principles

As mentioned in the state of the art, distributional approaches aim to calculate similarities between textual representations of word contexts. Methods to calculate similarities from IR seem then relevant for this problem. For a given word, the set of contexts of all its occurrences is considered as a document. The proximity between two words is then measured on their contexts by a similarity function from IR. This idea has many links with the work from the state of the art, but seems relatively unexplored, with the exeption of (Vechtomova and Robertson, 2012) in the specific context of similar named entities. It offers the advantage of being easily implementable because of the numerous IR tools available. Some adaptations are of course required. In contrast to IR, the stop words are kept as well as their positions relative to the considered occurrence. Lemmatization instead of stemming is performed. For example, in the excerpt: "... all forms of restrictions on freedom of expression, threats ...", the indexing terms restriction-2,

on-1, of+1, expression+2 are added to the context of freedom noted $\mathcal{C}(\text{freedom})$. The whole set of collected contexts for a word is used as a query in order to find its distributional neighbors. According to an IR similarity measure, the nearest words of this query (those whose contexts are closest) are returned.

We tested some of the most classical similarity measures used in IR: Hellinger (Escoffier, 1978; Domengès and Volle, 1979), TF-IDF/cosinus, and Okapi-BM-25 (Robertson et al., 1998). The last model can be seen as a variation of TF-IDF that better takes into account the difference between document sizes. This point is of importance since in our case the documents (namely the set of contexts of a word) are actually of very variable sizes, due to the very variable number of occurrences of each word. The Okapi-BM25 similarity between a word w_i ($\mathcal{C}(w_i)$ being the query), and w_j ($\mathcal{C}(w_j)$ being a document), is given in Eqn 1.

$$\text{similarity}(w_i, w_j) = \sum_{t \in \mathcal{C}(w_i)} \frac{(k_3+1) * qtf}{k_3 + qtf} * \frac{tf * (k_1+1)}{tf + k_1 * (1 - b + b * \frac{dl(\mathcal{C}(w_j))}{dl_{avg}})} * \log \frac{n - df(t) + 0.5}{df(t) + 0.5} \quad (1)$$

qtf is the number of occurrences of the word t in the query ($\mathcal{C}(w_i)$), dl is the size of $\mathcal{C}(w_j)$), dl_{avg} the average size of all contexts, n is the number of documents (that means in our case the number of considered words/thesaurus entries). $df(t)$ is the number of contexts ($\mathcal{C}(\cdot)$) containing t. Finally, k_1, k_3 and b are some constants, with default values $k_1 = 2$, $k_3 = 1000$ and $b = 0.75$. Details of these classical IR models are not given here but can be found in (Manning et al., 2008).

In the following experiments, the context of an occurrence is defined by the two words before and after the occurrence, and we also use an adjusted version of Okapi-BM25 similarity that enhances the influence of the document size and gives more importance to the most discriminating context words by setting $b = 1$ and putting the IDF squared to give more importance to the most discriminating context words.

3.2 Experimental setup

For the sake of comparison, we use in our experiments the data and baselines provided by Ferret (2013). The corpus used to build the distributional thesaurus is AQUAINT-2. It is a collection of articles from press containing about 380 million words. The thesaurus entries are all the nouns in the corpus with a frequency > 10. That represents 25,000 entries (i.e. unique nouns), denoted by n in the remaining. The corpus is labeled in parts of speech by TreeTagger (Schmid, 1994). In this way, we can identify the names that form the thesaurus entries and thus compare to existing work. However this information is not used to build the thesaurus, ensuring the portability of the method to other languages, similarly to (Freitag et al., 2005).

To evaluate the built thesauri, WordNet 3.0 synonyms (Miller, 1990) and Moby (Ward, 1996) are used as references, either separately, or jointly. These two resources exhibit quite different and complementary characteristics: on the one hand, WordNet indicates strong paradigmatic links between words (synonyms or quasi-synonyms). On the other hand, Moby groups words sharing more extended syntagmatic and paradigmatic relations, including synonymy, hyper/hypo-nymy, meronymy, but also many more complex types such as the composition of co-hyponymy and hyponymy (*abolition – annulment*, *cataclysm – debacle*) or hypernymy and co-hyponymy (*abyss – rift*, *algorithm – routine*). As a result, WordNet provides lists of 3 neighbors on average for the 10,473 names of the corpus it covers, while Moby provides lists of 50 neighbors on average for 9,216 names. When combined, the two resources provide a reference of 38 neighbors on average for 12,243 names. It is this combination of WordNet and Moby that will be used as the main reference in all evaluations of this article. Some results restricted to WordNet or Moby only as reference are also given in some cases to illustrate the impact of our methods on semantic simlarity versus semantic relatedness relations.

Through this intrinsic evaluation framework, the semantic neighbors of about half of the entries of our thesauri are evaluated, which can be considered as a very large evaluation set compared to classical benchmarks such as WordSim 353 for instance (Gabrilovich and Markovitch, 2007). This kind of intrinsic evaluation is of course limited by the relations that are present in the resources used as gold standards, often restricted to "classical" relation types such as synonymy or hypernymy. In our case, this limitation

Reference	Method	MAP	R-Prec	P@1	P@5	P@10	P@50	P@100
WordNet + Moby	Ferret 2013 *base*	5.6	7.7	22.5	14.1	10.8	5.3	3.8
	Ferret 2013 *best rerank*	6.1	8.4	24.8	15.4	11.7	5.7	3.8
	Hellinger	2.45	2.89	9.73	6.28	5.31	4.12	3.30
	TF-IDF	5.40	7.28	21.73	13.74	9.59	5.17	3.49
	Okapi-BM25	6.72	8.41	24.82	14.65	10.85	5.16	3.66
	Okapi-BM25 ajusted	8.97	10.94	31.05	18.44	13.76	6.46	4.54
	Ferret 2014 *synt*	7.9	10.7	29.4	18.9	14.6	7.3	5.2
WordNet	Ferret 2013 *base*	9.8	8.2	11.7	5.1	3.4	1.1	0.7
	Ferret 2013 *best rerank*	10.7	9.1	12.8	5.6	3.7	1.2	0.7
	Okapi-BM25 ajusted	14.17	12.22	16.97	7.10	4.47	1.41	0.84
	Ferret 2014 *synt*	13.3	11.5	15.6	6.9	4.5	1.5	0.9
Moby	Ferret 2013 *base*	3.2	6.7	24.1	16.4	13.0	6.6	4.8
	Ferret 2013 *best rerank*	3.5	7.2	26.5	17.9	14.0	6.9	4.8
	Okapi-BM25 ajusted	5.69	9.14	32.18	21.37	16.42	8.02	5.69
	Ferret 2014 *synt*	4.8	9.4	30.6	21.7	17.3	8.9	6.5

Table 1: Performance of IR models for distributional thesaurus building with the references WordNet, Moby and WordNet+Moby

holds true for WordNet's synonyms but can be considered as far less restrictive for the related words of Moby, due to the diversity of their underlying relation types.

3.3 Results

For a given name, our approach by IR models returns a list of names ordered by decreasing similarity. This list is compared to the reference one by computing several classical measures (expressed in % in the following): the precision after k first names, denoted P@k, the Mean Average Precision (MAP) which is the mean of the average precision scores for each query after a reference synonym is found, and the R-precision (precision at R-th position in the ranking of results, where R is the number of relevant names for the query).

Table 1 indicates the performance of different models of IR similarities. For purposes of comparison, we show the results obtained under the same conditions by Ferret (2013), with both a state of the art approach based on using cosine similarity over pointwise mutual information between contexts (referred as *base* in the table), and an improved version by learning as described in section 2 (referred as *best rerank*). We also give the results on the same corpus on an approach based on syntactic co-occurrents (Ferret, 2014 in press), extracted with the Minipar syntactic parser as in (Lin, 1998b).

In these early results, it is worth noting that some IR similarities are quite inefficient, including the TF alone or Hellinger similarity. This is hardly surprising since these similarities use very basic weights that do not enhance the discriminative contexts of words. The similarities that include a notion of IDF get better results in this. Okapi BM25-based similarities offer good results. The standard Okapi version yields performance similar to the state of the art, and the adjusted version even widely outperforms the two systems from Ferret (2013), in particular in terms of overall quality (measured by the MAP). Moreover, the results of this adjusted version are comparable to those obtained with syntactic co-occurrents while it only exploits window-based co-occurents, known to give usually worst results than syntactic co-occurrents, without even lemmatization. This latest version of the system serves as reference for the rest of this article.

4 Reciprocity in the graph of k-NN

Computing all the similarities between all pairs of words produces a weighted graph of neighbors: Each word is connected with certain strength to the n other words. The results above do not reflect this structure. The following sections aim to examine how take advantage of the neighborhood relations embedded in this graph. It must be first noted that some of the IR similarity measures we used are not symmetric, including Okapi-BM25. The similarity between a word w_i, used as query, and another word w_j does not give the same value as the similarity between the query w_j and w_i. Apart from that, even if

the similarity measure it-self is symmetric, nearest neighbor relationships are not.

It seems then reasonable to assume that the reciprocity between two adjacent words (each belonging to the k nearest neighbors of the other) is a sign of confidence on the proximity between these words. Using this information to improve the previous results is discussed in this section. In the following, $\tau_{w_i}(w_j)$ denotes the rank of the word w_j in the list of neighbors of w_i. $\tau_{w_i}(w_j)$ thus varies from 1 to n.

4.1 Distributional neighborhood graph

Reciprocal relationship in distributional neighborhood has already been discussed and used in some work (Ferret, 2013) on distributional semantic, or more generally, on nearest neighbors graphs (Pedronette et al., 2014). In these papers, the reciprocity was considered for giving a new similarity score in a simple way. For a word w_i and its neighbor w_j, the maximal or the minimal rank between $\tau_{w_i}(w_j)$ and $\tau_{w_j}(w_i)$ is taken as the new rank. These two operators have too severe effects as only one rank is taken into consideration to decide the final score. This leads to highly degraded performance as shown later. Many other aggregation operators have however been proposed in other contexts with a behavior may be more appropriate to the task, including fuzzy logic (Detyniecki, 2000). These operators carry some semantic that allow to comprehend their behavior, such as T-norms (fuzzy logic AND) and S-norms (or T-conorms, fuzzy OR).

In this section, we test some of these operators without claiming to be exhaustive. These are defined on $[0, 1]^2$, 1 being the certainty. They are used to generate a new similarity score according to:

$$\text{score}_{w_i}(w_j) = \text{Aggreg}(1 - \tau_{w_i}(w_j)/n, 1 - \tau_{w_j}(w_i)/n) \quad (2)$$

where Aggreg is an aggregation operator. The new scores are then used to produce a new list of nearest neighbors of w_i (the higher the score, the greater proximity is proven). We thus perceive the semantic associated with these operators. For example, if the aggregation function is $\max$, we get the expected behavior of the fuzzy OR associated with this S-norm: w_j will be ranked very close to w_i in the new list if w_j was close to w_i *or* if w_i was close to w_j. For the T-norm $\min$, this happens if w_j is close to w_i *and* w_i is close to w_j.

4.2 Results

Besides the $\min$ and $\max$ aggregation operators, Figure 1 reports the results obtained with the following T-norms (or T-norm families dependent on a parameter γ) used as aggregation function Aggreg:

$\text{T}_{\text{Prob}}(x, y) = x * y$

$\text{T}_{\text{Lukasiewicz}}(x, y) = \max(x + y - 1, 0)$

$\text{T}_{\text{Hamacher}}(x, y) = \frac{x*y}{\gamma+(1-\gamma)*(x+y-x*y)}$; $\gamma \geq 0$

$\text{T}_{\text{Yager}}(x, y) = \max(0, 1 - \sqrt[\gamma]{(1-x)^\gamma + (1-y)^\gamma})$; $\gamma > 0$

We also tested the standard related S-norms, obtained by generalization of the De Morgan's law: $S(x, y) = 1 - T(1 - x, 1 - y)$. For the T-norm families dependent on a parameter, we varied this parameter value in a systematic way. The results reported correspond to the parameter values that maximize the MAP.

All these operators get very different results. Some operators, such as min, max, Lukasiewicz, and others for some γ, induce a threshold effect which degrades the performance: they return a default value generating too much ex aequo among the neighbors, for some values of $\tau_{w_i}(w_j)$ and $\tau_{w_j}(w_i)$. T-norms, focusing on pairs of words symmetrically close to each other, are too restrictive. This is consistent with the conclusions of the work cited: if the reciprocity condition is applied too strictly, it does not improve the nearest neighbor lists over all the words. In contrast, S-norms seem better able to take advantage of the ranking. The improvements are modest in terms of overall quality (MAP), but important at some ranks (e.g. P@10).

Finally, it is important to note that these results depend heavily on the resource used as reference. We tested the aggregation rank with $\text{S}_{\text{Hamacher}}$, $\gamma = 0.95$, on Moby and WordNet references separately. Results are given in Table 2. Because Wordnet is based on a synonymy relationship strong enough (and therefore reciprocal), the performance gains on WordNet are much higher than on Moby.

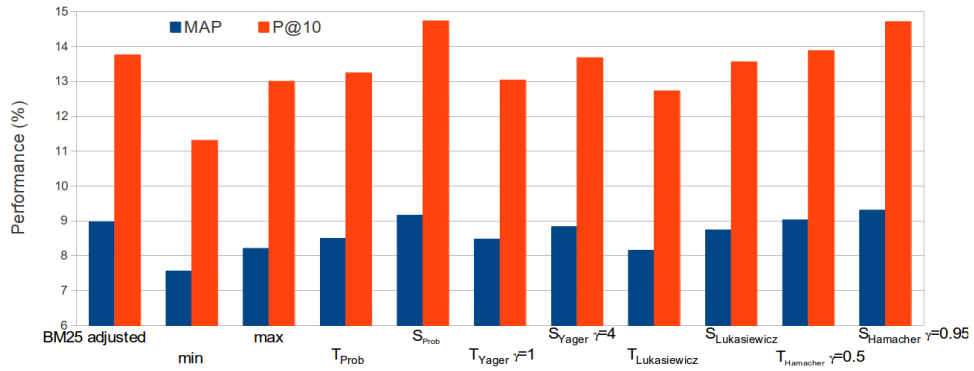


Figure 1: Performance of reciprocal rank aggregation, on the reference WordNet+Moby

Reference	MAP	R-Prec	P@1	P@5	P@10	P@50	P@100
WordNet + Moby	9.30 (+3.75)	11.06 (+2.03)	30.42 (-2.53)	19.29 (+4.58)	14.71 (+6.92)	7.09 (+9.78)	4.86 (+7.07)
WordNet	15.05 (+6.23)	12.81 (+4.81)	17.55 (+3.41)	7.96 (+12.16)	5.07 (+13.30)	1.63 (+15.69)	0.94 (+12.23)
Moby	5.90 (+3.65)	11.86 (+4.14)	*31.77 (-1.27)*	21.65 (+1.34)	17.0 (+3.53)	8.42 (+5.01)	5.92 (+4.12)

Table 2: Performance and gains (%) of reciprocal rank aggregation, relatively to adjusted Okapi-BM25, on the references WordNet and Moby taken separately, with aggregation operator S_{Hamacher}, $\gamma = 0.95$

5 Confidence estimation for a distributional neighborhood list

In the previous section, the rank of w_i in the list of neighbors of w_j is used to improve the ranking of w_j in the list of neighbors of w_i. We can also be interested in a more general way to the relative positions of w_i and w_j in all neighbor lists of all the words. Thereby, we expect to derive a more complete information. As a first step, we define a confidence criterion associated with each list of nearest neighbors, only based on the neighborhood graph.

5.1 Principle

We make the following assumption: the nearest neighbor list of a word w is probably of good quality if the distance (in terms of rank) between w and each of its neighbors w_i, denoted $\delta(w, w_i)$, is consistent with the distance observed between these same words (w, w_i) in other lists. The intuition here is that words supposed to be close should also be found close to the same other words. If k nearest neighbors of w have this property, then we attribute a high confidence to the neighbor list of w.

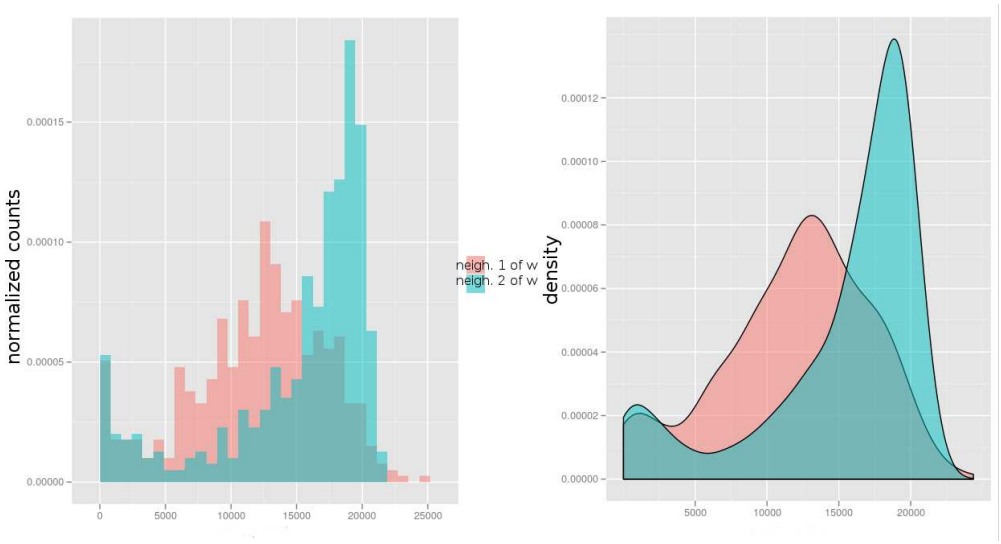
Formally, we define the confidence of the k-nearest neighbor list of w by:

$$Q(w) = \prod_{\{w_i | \tau_w(w_i) \leq k\}} p(\delta(w, w_i) = \tau_w(w_i)) \tag{3}$$

where $p(\delta(w, w_i) = \tau_w(w_i))$ is the probability that w_i is the $\tau_w(w_i)$-th neighbor of w. The problem is then to estimate the probability distribution $p(\delta(w, w_i))$ for each pair of words (w, w_i). To achieve this goal, we use the Parzen windows which is a method for nonparametric density estimation. We describe below how this classic method (Parzen, 1962; Wasserman, 2005) is applied in our context.

5.2 Parzen-window density estimation

Let $x_{ab} = \delta(w_a, w_b)$ be the distance (in terms of ranks) between two words w_a and w_b in a list of neighbors of any given word. Considering the n words of the thesaurus, we have a sample of n realizations assumed *iid*: (x_{ab}^1, x_{ab}^2, ... , x_{ab}^n), which are the observed distances between w_a et w_b in each (complete) neighbor list of each word. These counts can be represented by an histogram as illustrated in Figure 2 (a). Using the Parzen window technique, we can then estimate the probability density of x_{ab} with a kernel



(a) (b)

Figure 2: (a) Example of two distributions of distances x_{ab} and x_{ac} between a word w_a and two of its neighbors w_b and w_c, represented as histograms (blue and red) (b) Same distributions represented by densities estimated with the Parzen-windows method.

density estimator with Eqn 4 where h is a smoothing parameter called the bandwidth, and $K(\dot{})$ is a kernel that we choose Gaussian. The resulting density is illustrated in Figure 2 (b).

$$\widehat{p_h}(x_{ab}) = \frac{1}{n\,h}\sum_{i=1}^{n} K\left(\frac{x_{ab} - x_{ab}^i}{h}\right) \quad \text{with} \quad K(u) = \frac{1}{\sqrt{2\pi}} e^{\frac{-u^2}{2}} \tag{4}$$

Thus, the resulting probability is a mixture of Gaussians centered on each x_{ab}^i. These methods are known to be sensitive to the bandwidth h, which controls the regularity of the estimation. The problem of choosing h is crucial in density estimation and was widely discussed in the literature. We use Silverman's rule of thumb (Silverman, 1986, page 48, eqn (3.31)) to set its value. Under the assumption of normality of the underlying distribution, this rule provides a simple way to calculate the optimal parameter h when Gaussian functions are used to approximate univariate data (Eqn 5 where $\hat{\sigma}$ is the standard deviation of the samples, and $q_3 - q_1$ is the interquartile range).

$$\hat{h} = 0.9\ \min(\hat{\sigma}, \frac{q_3 - q_1}{1.34})\ n^{-\frac{1}{5}} \tag{5}$$

Once these probabilities have been estimated on each of the k-nearest neighbors of w, we can calculate the confidence score $Q(w)$. The complexity of this estimation for all neighbor lists is $\mathcal{O}(k * n^2)$.

5.3 Using the confidence score

The expected benefit of using the confidence score is to have an a priori indication on the quality of a neighbor list for a given word. Such a score may thus be useful for many applications using thesauri produced by our approach (e.g. for expanding queries in information retrieval tasks). An evaluation of the confidence score through such applications would certainly be the most suitable, but beyond the scope of this article. We use default direct assessment towards the MAP: we measure the correlation between MAP and the confidence score, the idea being that an entry with a neighbor list of low quality matches an entry with low MAP. We use Spearman's correlation ρ and the Kendall's rank correlation coefficient τ, which do not make any assumption about linearity and compare only the order of words classified according to their MAP with the order according to their confidence score. The results of these coefficients are given in Table 3, along with p-value of the associated test of significance. A coefficient

Correlation coefficient	value	statistical significance
Kendall τ	0.37	$p < 10^{-64}$
Spearman ρ	0.51	$p < 10^{-64}$

Table 3: Correlation coefficient values between the MAP and the confidence score, and their statistical significance (p-value).

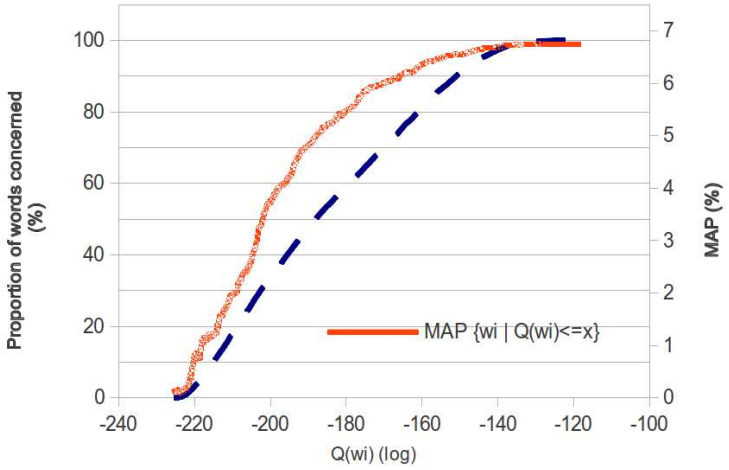


Figure 3: Average MAP computed on words with a confidence score lower than a threshold q (x-axis, log-scale), and cumulative proportion of concerned words.

value of 1 indicates a perfect correlation, 0 no correlation and -1 an inverse correlation. A low p-value, for example < 0.05, indicates a statistically significant result. The confidence scores are obtained with $k = 20$. Other experiments, not reported here, show that this parameter k has little influence on the correlation, for values between 5 and 100.

These measures attest to some statistically significant correlation between our confidence score and the MAP, however this correlation is imperfect and non-linear. We compute the average MAP on neighbor lists with a confidence score lower than a threshold q. Figure 3 represents the average MAP (y-axis) in function of the threshold q (x-axis). It shows that the confidence score is still a good indicator of quality, as the MAP decreases with the confidence score.

The confidence score can be used to improve the performance of aggregation techniques presented in Section 4 by integrating it in the final score:

$$\text{score}_{w_i}(w_j) = Q(w_j) * \text{Aggreg}(1 - \tau_{w_i}(w_j)/n, 1 - \tau_{w_j}(w_i)/n) \quad (6)$$

As shown in Table 4, using this information allows even greater gains than those reported in the previous section (a Wilcoxon test ($p < 0.05$) (Hull, 1993) is performed to ensure that the differences are statistically significant; non-significant ones are shown in italics). In the next section, we propose another use of the confidence scores to improve results more specifically on the head of the lists, that is to say on the neighbors judged closest.

Method	MAP	R-Prec	P@1	P@5	P@10	P@50	P@100
S_{Hamacher} $\gamma = 0.95$	9.61 (+7.20)	11.59 (+5.85)	*30.86 (-0.53)*	19.52 (+5.83)	14.76 (+7.24)	7.03 (+8.88)	4.93 (+8.67)

Table 4: Performance gains (%) by reciprocal rank aggregation using the confidence score, on WordNet+Moby reference.

Target	MAP	R-Prec	P@1	P@5	P@10
all words	9.16 (+2.17)	11.24 (+2.76)	*30.73 (-1.02)*	19.30 (+4.64)	14.37 (+4.44)
the third of words with the lowest $Q(w)$	9.55 (+6.44)	11.81 (+7.99)	31.85 (+2.56)	20.43 (+10.81)	15.46 (+12.37)

Table 5: Performance gains (%) of reranking with the Hungarian algorithm.

6 Local reranking

The previous method gives an overall score to the list, but one can also make use of the individual ranking probabilities $p(\delta(w_i, w_j))$, estimated according to the method of Parzen windows. For a given word w, we have for each of its neighbors w_j the probability of his current rank: $p(\delta(w, w_j)) = \tau_w(w_j)$. For a given neighbor w_j, we can also calculate the probability of any other rank τ: $p(\delta(w, w_j)) = \tau$ with $\tau = 1, 2, ...$ In this section, we propose to rely on these more local information to improve the performance by reranking the k-nearest neighbors.

6.1 Reranking by the Hungarian algorithm

A simple approach would be to reorder the list based on this criterion, from the most probable neighbors to the least ones. But ranking probability estimation for each word is imperfect, and such a reranking strongly degrades the results. We therefore propose instead a method to rerank the k-nearest neighbors on a more local and controlled manner: a word that was not originally in the k-nearest neighbors can not become a k-nearest neighbor, and a word can not be reranked too far from its original rank.

Our problem is expressed by the following matrix $\mathcal{M}_{\text{profit}}$. The rows correspond to words in their original ranks (denoted w_1 to w_k), the columns to new ranks τ at which these words can be assigned, and matrix values are the probabilities of each word w_j to appear at rank τ. Given these probabilities, the goal is to find the most likely permutation of the k-nearest neighbors.

$$\mathcal{M}_{\text{profit}} = \begin{pmatrix} p(\delta(w, w_1) = 1) & \cdots & p(\delta(w, w_1) = k) \\ \vdots & \ddots & \vdots \\ p(\delta(w, w_k) = 1) & \cdots & p(\delta(w, w_k) = k) \end{pmatrix}$$

As pointed out, we want to avoid that an initially very close neighbor was moved far away and vice versa. This constraint is added by multiplying the matrix $\mathcal{M}_{\text{profit}}$ by a penalty matrix $\mathcal{M}_{\text{penalty}}$ (see below) with the Hadamard product (element by element matrix product, denoted $\circ$).

$$\mathcal{M}_{\text{penalty}} = \begin{pmatrix} 1 & \frac{k-1}{k} & \cdots & 0 \\ \frac{k-1}{k} & 1 & \cdots & \frac{1}{k} \\ \vdots & \ddots & \vdots & \\ 0 & \frac{1}{k} & \cdots & 1 \end{pmatrix}$$

We then face a combinatorial optimization problem which can be solved in polynomial time by the Hungarian method (Kuhn and Yaw, 1955, for a description of the algorithm) on the matrix of assignment costs $\mathcal{M}_{\text{profit}} \circ \mathcal{M}_{\text{penalty}}$. This algorithm was originally proposed to optimize the assignment of workers (in our case, the neighbors) on tasks (in our case, ranks), according to the profit generated by each worker for each task (in our case, the probability that a neighbor stands at a given rank). The result of this algorithm therefore indicates a new rank for each word. The algorithm converges to an optimal solution with a complexity $\mathcal{O}(k^3)$ (for reranking the k-nearest neighbors).

6.2 Results

Table 5 presents the performance achieved by our local reranking method compared to the adjusted Okapi-BM25 reference using the same experimental conditions as above. As before, the considered neighborhood is set to $k = 20$. Precisions beyond this threshold are unchanged and thus not reported. We test the effectiveness of the local reranking on all neighbor lists and on a third of lists with the lowest quality scores.

It appears that the reranking on the whole lists does not provide a real gain. However, the gain is substantial on the lists with low confidence score. Moreover, unlike the experiments of section 4, these

gains apply by construction to the heads of lists, which are most likely to be used in practice. This difference between results on the whole set of words and on those with the lowest confidence scores can be explained in two ways. First, the lists with the highest confidence scores correspond largely to the lists with the best MAP, as expected and illustrated in Figure 3. This therefore suggests a priori little room for improvement. Second, regardless of MAP, we can also assume that these lists already have an optimum arrangement of individual probabilities that explains the high confidence score. The reranking thus concerns only few neighbors.

7 Conclusion and future work

The different contributions proposed in this article do not place themselves all at the same level. The thesaurus construction using tools from the IR is not a major conceptual innovation, but this approach seems curiously unexplored although it provides very competitive results while requiring minimum implementations through existing tools from IR.

The various propositions exploiting the neighborhood graph to improve the thesaurus are part of a more original approach where the whole thesaurus is considered. We have specifically examined the aspects of reciprocity and distance, in terms of rank, between two words to offer several contributions. The improvements obtained by aggregation over all neighbors or by the local reranking from confidence scores validate our approach. It should be noted that the gains are small in absolute terms, but, compared to those observed in the field, correspond to significant improvements.

The various aspects of this work open up many prospects of research. For example, many other aggregate functions in addition to those tested in section 4 exist in the literature. Some may even offer the possibility of integrating the confidence score associated with each neighbor, as Choquet's or Sugeno's integrals (Detyniecki, 2000). More generally, it would be interesting to iteratively use improvements of neighbor lists to update the confidence scores, etc., in the spirit for example of what is proposed by Pedronette et al. (2014). A detailled analysis of the impact of these techniques according to the type of semantic relation is still to be performed. Beyond the distributional thesauri construction, the proposed methods to compute confidence scores or reorder lists of neighbors can be applied to other problems where the k-nearest neighbor graphs of are built. Also note that we have only considered a small part of the information carried by the neighborhood graph. We focused on the aspects of reciprocity, but taking into account other aspects of the graph (in particular the transitivity, or more generally its topology), could lead to further improvements.

References

Clémentine Adam, Cécile Fabre, and Philippe Muller. 2013. Évaluer et améliorer une ressource distributionnelle : protocole d'annotation de liens sémantiques en contexte. *TAL*, 54(1):71–97.

Bartosz Broda, Maciej Piasecki, and Stan Szpakowicz. 2009. Rank-Based Transformation in Measuring Semantic Relatedness. In *22nd Canadian Conference on Artificial Intelligence*, pages 187–190.

Alexander Budanitsky and Graeme Hirst. 2006. Evaluating WordNet-based measures of lexical semantic relatedness. *Computational Linguistics*, 32(1):13–47.

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.

Marcin Detyniecki. 2000. *Mathematical aggregation operators and their application to video querying*. Ph.D. thesis, Université de Paris 6.

Dominique Domengès and Michel Volle. 1979. Analyse factorielle sphérique : une exploration. *Annales de l'INSEE*, 35:3–83.

Bernard Escoffier. 1978. Analyse factorielle et distances répondant au principe d'équivalence distributionnelle. *Revue de statistique appliquée*, 26(4):29–37.

Olivier Ferret. 2012. Combining bootstrapping and feature selection for improving a distributional thesaurus. In *20th European Conference on Artificial Intelligence (ECAI 2012)*, pages 336–341, Montpellier, France.

Olivier Ferret. 2013. Identifying bad semantic neighbors for improving distributional thesauri. In *51st Annual Meeting of the Association for Computational Linguistics (ACL 2013)*, pages 561–571, Sofia, Bulgaria.

Olivier Ferret. 2014 (in press). Typing relations in distributional thesauri. In N. Gala, R. Rapp, and G. Bel, editors, *Advances in Language Production, Cognition and the Lexicon*. Springer.

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

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

Evgeniy Gabrilovich and Shaul Markovitch. 2007. Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *20th International Joint Conference on Artificial Intelligence (IJCAI 2007)*, pages 6–12.

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

Masato Hagiwara, Yasuhiro Ogawa, and Katsuhiko Toyama. 2006. Selection of effective contextual information for automatic synonym acquisition. In *21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (COLING-ACL 2006)*, pages 353–360, Sydney, Australia.

Eric H. Huang, Richard Socher, Christopher D. Manning, and Andrew Y. Ng. 2012. Improving word representations via global context and multiple word prototypes. In *50th Annual Meeting of the Association for Computational Linguistics (ACL'12)*, pages 873–882.

David Hull. 1993. Using Statistical Testing in the Evaluation of Retrieval Experiments. In *Proc. of the 16th Annual ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'93*, Pittsburgh, États-Unis.

Harold W. Kuhn and Bryn Yaw. 1955. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97.

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.

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

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

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013)*, pages 746–751, Atlanta, Georgia.

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

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

Emanuel Parzen. 1962. On estimation of a probability density function and mode. *Ann. Math. Stat.*, 33:1065–1076.

Daniel Carlos Guimarães Pedronette, Otávio Augusto Bizetto Penatti, and Ricardo da Silva Torres. 2014. Unsupervised manifold learning using reciprocal knn graphs in image re-ranking and rank aggregation tasks. *Image Vision Computing*, 32(2):120–130.

Stephen E. Robertson, Steve Walker, and Micheline Hancock-Beaulieu. 1998. Okapi at TREC-7: Automatic Ad Hoc, Filtering, VLC and Interactive. In *Proc. of the 7th Text Retrieval Conference, TREC-7*, pages 199–210.

Magnus Sahlgren. 2001. Vector-based semantic analysis: Representing word meanings based on random labels. In *ESSLLI 2001 Workshop on Semantic Knowledge Acquisition and Categorisation*, Helsinki, Finland.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*.

Bernard W. Silverman. 1986. *Density estimation for statistics and data analysis*. Monographs on statistics and applied probability. Chapman and Hall Boca Raton, London, Glasgow, Weinheim.

Tim Van de Cruys. 2010. *Mining for Meaning. The Extraction of Lexico-semantic Knowledge from Text*. Ph.D. thesis, University of Groningen, The Netherlands.

Olga Vechtomova and Stephen E. Robertson. 2012. A domain-independent approach to finding related entities. *Information Processing and Management*, 48(4):654–670.

Grady Ward. 1996. Moby thesaurus. Moby Project.

Larry Wasserman. 2005. *All of Statistics: A Concise Course in Statistical Inference*. Springer Texts in Statistics.

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

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