Synonym Extraction Using a Semantic Distance on a Dictionary

Philippe Muller
IRIT – CNRS, UPS & INPT
Toulouse, France
`muller@irit.fr`

Nabil Hathout
ERSS – CNRS & UTM
Toulouse, France
`hathout@univ-tlse2.fr`

Bruno Gaume
IRIT – CNRS, UPS & INPT
Toulouse, France
`gaume@irit.fr`

Abstract

Synonyms extraction is a difficult task to achieve and evaluate. Some studies have tried to exploit general dictionaries for that purpose, seeing them as graphs where words are related by the definition they appear in, in a complex network of an arguably semantic nature. The advantage of using a general dictionary lies in the coverage, and the availability of such resources, in general and also in specialised domains. We present here a method exploiting such a graph structure to compute a distance between words. This distance is used to isolate candidate synonyms for a given word. We present an evaluation of the relevance of the candidates on a sample of the lexicon.

1 Introduction

Thesaurus are an important resource in many natural language processing tasks. They are used to help information retrieval (Zukerman et al., 2003), machine or semi-automated translation, (Ploux and Ji, 2003; Barzilay and McKeown, 2001; Edmonds and Hirst, 2002) or generation (Langkilde and Knight, 1998). Since the gathering of such lexical information is a delicate and time-consuming endeavour, some effort has been devoted to the automatic building of sets of synonyms words or expressions.

Synonym extraction suffers from a variety of methodological problems, however. Synonymy itself is not an easily definable notion. Totally equivalent words (in meaning and use) arguably do not exist, and some people prefer to talk about near-synonyms (Edmonds and Hirst, 2002). A near-synonym is a word that can be used instead of another one, in some contexts, without *too much* change in meaning. This leaves of lot of freedom in the degree of synonymy one is ready to accept. Other authors include "related" terms in the building of thesaurus, such as hyponyms and hypernyms, (Blondel et al., 2004) in a somewhat arbitrary way. More generally, *paraphrase* is a preferred term referring to alternative formulations of words or expressions, in the context of information retrieval or machine translation.

Then there is the question of evaluating the results. Comparing to already existing thesaurus is a debatable means when automatic construction is supposed to complement an existing one, or when a specific domain is targeted, or when simply the automatic procedure is supposed to fill a void. Manual verification of a sample of synonyms extracted is a common practice, either by the authors of a study or by independent lexicographers. This of course does not solve problems related to the definition of synonymy in the "manual" design of a thesaurus, but can help evaluate the relevance of synonyms extracted automatically, and which could have been forgotten. One can hope at best for a semi-automatic procedure were lexicographers have to weed out bad candidates in a set of proposals that is hopefully not too noisy.

A few studies have tried to use the lexical information available in a general dictionary and find patterns that would indicate synonymy relations (Blon-

del et al., 2004; Ho and Cédrick, 2004). The general idea is that words are related by the definition they appear in, in a complex network that must be semantic in nature (this has been also applied to word sense disambiguation, albeit with limited success (Veronis and Ide, 1990; H.Kozima and Furugori, 1993)).

We present here a method exploiting the graph structure of a dictionary, where words are related by the definition they appear in, to compute a distance between words. This distance is used to isolate candidate synonyms for a given word. We present an evaluation of the relevance of the candidates on a sample of the lexicon.

2 Semantic distance on a dictionary graph

We describe here our method (dubbed Prox) to compute a distance between nodes in a graph. Basically, nodes are derived from entries in the dictionary or words appearing in definitions, and there are edges between an entry and the word in its definition (more in section 3). Such graphs are "small world" networks with distinguishing features and we hypothetize these features reflect a linguistic and semantic organisation that can be exploited (Gaume et al., 2005).

The idea is to see a graph as a Markov chain whose states are the graph nodes and whose transitions are its edges, valuated with probabilities. Then we send random particles walking through this graph, and their trajectories and the dynamics of their trajectories reveal their structural properties. In short, we assume the average distance a particle has made between two nodes after a given time is an indication of the semantic distance between these nodes. Obviously, nodes located in highly clustered areas will tend to be separated by smaller distance.

Formally, if $G = (V, E)$ is a reflexive graph (each node is connected to itself) with $|V| = n$, we note $[G]$ the $n \times n$ adjacency matrix of G that is such that $[G]_{i,j}$ (the ith row and j^{th} column) is non null if there is an edge between node i and node j and 0 otherwise. We can have different weights for the edge between nodes (cf. next section), but the method will be similar.

The first step is to turn the matrix into a Markovian matrix. We note $[\hat{G}]$ the Markovian matrix of G,

such that

$$[\hat{G}]_{r,s} = \frac{[G]_{r,s}}{\sum_{x \in V}([G]_{r,x})}$$

The sum of each line of G is different from 0 since the graph is reflexive.

We note $[\hat{G}]^i$ the matrix $[\hat{G}]$ multiplied i times by itself.

Let now $\text{PROX}(G, i, r, s)$ be $[\hat{G}]^i_{r,s}$. This is thus the probability that a random particle leaving node r will be in node s after i time steps. This is the measure we will use to determine if a node s is closer to a node r than another node t. The choice for i will depend on the graph and is explained later (cf. section 4).

3 Synonym extraction

We used for the experiment the XML tagged MRD *Trésor de la Langue Française informatisé* (TLFi) from ATILF (http://atilf.atilf.fr/), a large French dictionary with 54,280 articles, 92,997 entries and 271,166 definitions. The extraction of synonyms has been carried out only for nouns, verbs and adjectives. The basic assumption is that words with semantically close definitions are likely to be synonyms. We then designed a oriented graph that brings closer definitions that contain the same words, especially when these words occur in the beginning. We selected the noun, verb and adjective definitions from the dictionary and created a record for each of them with the information relevant to the building of the graph: the word or expression being defined (henceforth, *definiendum*); its grammatical category; the hierarchical position of the defined (sub-)sense in the article; the definition proper (henceforth *definiens*).

Definitions are made of 2 members: a *definiendum* and a *definiens* and we strongly distinguish these 2 types of objects in the graph. They are represented by 2 types of nodes: a-type nodes for the words being defined and for their sub-senses; o-type nodes for the words that occur in *definiens*.

For instance, the noun *nostalgie* 'nostalgia' has 6 defined sub-senses numbered A.1, A.2, B., C., C. – and D.:

NOSTALGIE, subst. fém.
A. 1. État de tristesse [...]
2. Trouble psychique [...]
B. Regret mélancolique [...] désir d'un retour dans le passé.
C. Regret mélancolique [...] désir insatisfait.
– Sentiment d'impuissance [...]
D. État de mélancolie [...]

The 6 sub-senses yield 6 a-nodes in the graph plus one for the article entry:

```
a.S.nostalgie        article entry
a.S.nostalgie.1_1    sub-sense A. 1.
a.S.nostalgie.1_2    sub-sense A. 2.
a.S.nostalgie.2      sub-sense B.
a.S.nostalgie.3      sub-sense C.
a.S.nostalgie.3_1    sub-sense C. –
a.S.nostalgie.4      sub-sense D.
```

A-node tags have 4 fields: the node type (namely `a`); its grammatical category (S for nouns, V for verbs and A for adjectives); the lemma that correponds to the *definiendum*; a representation of the hierarchical position of the sub-sense in the dictionary article. For instance, the **A. 2.** sub-sense of *nostalgie* corresponds to the hierarchical position 1_2.
O-nodes represent the types that occur in *definiens*.[1] A second example can be used to present them. The adjective *jonceux* 'rushy' has two sub-senses 're-sembling rushes' and 'populated with rushes':

Jonceux, -euse,
a) Qui ressemble au jonc.
b) Peuplé de joncs.

Actually, TLFi definitions are POS-tagged and lem-matized:

Jonceux/S
a) qui/Pro ressembler/V au/D jonc/S ./X
b) peuplé/A de/Prep jonc/S ./X [2]

The 2 *definiens* yield the following o-type nodes in the graph:
`o.Pro.qui; o.V.ressembler; o.D.au; o.S.jonc; o.X..; o.A.peuplé; o.Prep.de`

---

[1]The tokens are represented by edges.
[2]In this sentence, *peuplé* is an adjective and not a verb.

All the types that occur in *definiens* are represented, including the function words (pronouns, deter-miners...) and the punctuation. Function words play an important role in the graph because they bring closer the words that belong to the same semantical referential classes (e.g. the adjectives of resemblance), that is words that are likely to be synonyms. Their role is also reinforced by the manner edges are weighted.
A large number of TLFi definitions concerns phrases and locutions. However, these definitions have been removed from the graph because:

- their tokens are not identified in the *definiens*;

- their grammatical categories are not given in the articles and are difficult to calculate;

- many lexicalized phrases are not sub-senses of the article entry.

O-node tags have 3 fields: the node type (namely `o`); the grammatical category of the word; its lemma.
The oriented graph built for the experiment then contains one a-node for each entry and each entry sub-sense (i.e. each *definiendum*) and one o-node for each type that occurs in a definition (i.e. in a *definiens*). These nodes are connected as follows:

1. The graph is reflexive;

2. Sub-senses are connected to the words of their *definiens* and vice versa (e.g. there is an edge between `a.A.jonceux.1` and `o.Pro.qui`, and another one between `o.Pro.qui` and `a.A.jonceux.1`).

3. Each a-node is connected to the a-nodes of the immediately lower hierarchical level but there is no edge between an a-node and the a-nodes of higher hier-archical levels (e.g. `a.S.nostalgie` is connected to `a.S.nostalgie.1_1`, `a.S.nostalgie.1_2`, `a.S.nostalgie.2`, `a.S.nostalgie.3` and `a.S.nostalgie.4`, but none of the sub-senses is connected to the entry).

4. Each o-node is connected to the a-node that represents its entry, but there is no edge between the a-node representing an entry and the corresponding o-node (e.g. there is an edge between `o.A.jonceux` and `a.A.jonceux`, but none between `a.A.jonceux` and `o.A.jonceux`).

All edge weights are 1 with the exception of the edges representing the 9 first words of each *definiens*. For these words, the edge weight takes into account their position in the *definiens*. The weight of the edge that represent the first token is 10; it is 9 for the second word; and so on down to 1.[3]

These characteristics are illustrated by the fragment of the graph representing the entry *jonceux* in table 1.

4 Experiment and results

Once the graph built, we used Prox to compute a semantic similarity between the nodes. We first turned the matrix G that represent the graph into a Markovian matrix $[\hat{G}]$ as described in section 2 and then computed $[\hat{G}]^5$, that correspond to 5-steps paths in the Markovian graph.[4] For a given word, we have extracted as candidate synonyms the a-nodes (*i*) of the same category as the word (*ii*) that are the closest to the o-node representing that word in the dictionary definitions. Moreover, only the first a-node of each entry is considered. For instance, the candidate synonyms of the verb *accumuler* 'accumulate' are the a-nodes representing verbs (i.e. their tags begin in `a.V`) that are the closer to the `o.V.accumuler` node.

5-steps paths starting from an o-node representing a word w reach six groups of a-nodes:

$\mathcal{A}_1$ the a-nodes of the sub-senses which have w in their definition;

$\mathcal{A}_2$ the a-nodes of the sub-senses with *definiens* containing the same words as those of $\mathcal{A}_1$;

$\mathcal{A}_3$ the a-nodes of the sub-senses with *definiens* containing the same words as those of $\mathcal{A}_2$;

$\mathcal{B}_1$ the a-nodes of the sub-senses of the article of w. (These dummy candidates are not kept.)

$\mathcal{B}_2$ the a-nodes of the sub-senses with *definiens* containing the same words as those of $\mathcal{B}_1$;

$\mathcal{B}_3$ the a-nodes of the sub-senses with *definiens* containing the same words as those of $\mathcal{B}_2$;

The three first groups take advantage of the fact that synonyms of the *definiendum* are often used in definiens.

The question of the evaluation of the extraction of synonyms is a difficult one, as was already mentioned in the introduction. We have at our disposal several thesauri for French, with various coverages (from about 2000 pairs of synonyms, to 140,000), and a lot of discrepancies.[5] If we compare the thesaurus with each other and restrict the comparison to their common lexicon for fairness, we still have a lot of differences. The best f-score is never above 60%, and it raises the question of the proper gold standard to begin with. This is all the more distressing as the dictionary we used has a larger lexicon than all the thesaurus considered together (roughly twice as much). As our main purpose is to build a set of synonyms from the TLF to go beyond the available thesaurus, we have no other way but to have lexicographers look at the result and judge the quality of candidate synonyms. Before imposing this workload on our lexicographer colleagues, we took a sample of 50 verbs and 50 nouns, and evaluated the first ten candidates for each, using the ranking method presented above, and a simpler version with equal weights and no distinction between sense levels or node types. The basic version of the graph also excludes nodes with too many neighbours, such as "être" (be), "avoir" (have), "chose" (thing), etc.). Two of the authors separately evaluated the candidates, with the synonyms from the existing thesauri

---

[3]Lexicographic definitions usually have two parts: a *genus* and a *differentia*. This edge weight is intended to favour the *genus* part of the *definiens*.

[4]The path length has been determined empirically.

[5]These seven classical dictionaries of synonyms are all available from http://www.crisco.unicaen.fr/dicosyn.html.

	o.A.jonceux	a.A.jonceux	a.A.jonceux.1	a.A.jonceux.2	o.Pro.qui	o.V.ressembler	o.D.au	o.S.jonc	o.X..	o.A.peuplé	o.Prep.de
o.A.jonceux	1	1									
a.A.jonceux		1	1	1							
a.A.jonceux.1			1		1	1	1	1	1		
a.A.jonceux.2				1				1	1	1	1
o.Pro.qui			10		1						
o.V.ressembler			9			1					
o.D.au			8				1				
o.S.jonc			7	8				1			
o.X..			6	7					1		
o.A.peuplé				10						1	
o.Prep.de				9							1

Table 1: A fragment of the graph, presented as a matrix.

already marked. It turned out one of the judge was much more liberal than the other about synonymy, but most synonyms accepted by the first were accepted by the second judge (precision of 0.85).[6]

We also considered a few baselines inspired by the method. Obviously a lot of synonyms appear in the definition of a word, and words in a definition tend to be consider close to the entry they appear in. So we tried two different baselines to estimate this bias, and how our method improves or not from this.

The first baseline considers as synonyms of a word all the words of the same category (verbs or nouns in each case) that appear in a definition of the word, and all the entry the word appear in. Then we selected ten words at random among this base.

The second baseline was similar, but restricted to the *first* word appearing in a definition of another word. Again we took ten words at random in this set if it was larger than ten, and all of them otherwise.

We show the results of precision for the first candidate ranked by prox, the first 5, and the first 10 (always excluding the word itself). In the case of the two baselines, results for the first ten are a bit

---

misleading, since the average numbers of candidates proposed by these methods were respectively 8 and 6 for verbs and 9 and 5.6 for nouns (Table 2). Also, nouns had an average of 5.8 synonyms in the existing thesauri (when what was considered was the min between 10 and the number of synonyms), and verbs had an average of 8.9.

We can see that both baselines outperforms weighted prox on the existing thesaurus for verbs, and that the simpler prox is similar to baseline 2 (first word only). For nouns, results are close between B2 and the two proxs. It is to be noted that a lot of uncommon words appear as candidates, as they are related with very few words, and a lot of these do not appear in the existing thesauri.

By looking precisely at each candidate (see judges' scores), we can see that both baselines are slightly improved (and still close to one another), but are now beaten by both prox for the first and the first 5 words. There is a big difference between the two judges, so Judge 2 has better scores than Judge 1 for the baselines, but in each case, prox was better. It could be troubling to see how good the second baseline is for the first 10 candidates, but one must remember this baseline actually proposes 6 candidates on average (when prox was always at 10), making it actually nothing more than a variation on the 5

		Existing Thesauri (V)	Judge 1	Judge 2	ET (N)	J1	J2
baseline-1	1	0.30	0.42	0.38	0.06	0.12	0.12
	5	0.29	0.39	0.375	0.08	0.12	0.13
	10	0.31	0.41	0.39	0.10	0.14	0.15
baseline-2	1	0.32	0.52	0.44	0.21	0.22	0.23
	5	0.36	0.50	0.446	0.21	0.24	0.25
	10	0.28	0.51	0.46	0.19	0.245	0.255
simple prox	1	0.35	0.67	NA	0.27	0.415	0.417
	5	0.34	0.52	NA	0.137	0.215	0.237
	10	0.247	0.375	NA	0.123	0.17	0.19
weighted prox	1	0.22	0.56	0.76	0.18	0.44	0.5
	5	0.196	0.44	0.58	0.148	0.31	0.39
	10	0.17	0.36	0.47	0.10	0.22	0.3

Table 2: Experimental results on a sample, V=verbs, N=nouns,

candidate baseline, to which it should be compared in all fairness (and we see that prox is much better there). The difference between the two versions of prox shows that a basic version is better for verbs and the more elaborate one is better for nouns, with overall better results for verbs than for nouns.

One could wonder why there was some many more candidates marked as synonyms by both judges, compared to the original compilation of thesaurus. Mainly, it seemed to us that it can be accounted for by a lot of infrequent words, or old senses of words absent for more restricted dictionaries. We are currently investigating this matter. It could also be that our sample picked out a lot of not so frequent words since they outnumber frequent words in such a large dictionary as the TLF. An indication is the average frequency of words in a corpus of ten years of the journal "Le Monde". The 50 words picked out in our sample have an average frequency of 2000 occurrences, while when we consider all our about 430 candidates for synonymy, the average frequency is 5300.

The main conclusion to draw here is that our method is able to recover a lot of synonyms that are in the definition of words, and some in definitions not directly related, which seems to be an improvement on previous attempts from dictionaries. There is some arbitrariness in the method that should be further investigated (the length of the random walk for instance), but we believe the parameters are rather intuitive wrt to graph concepts. We also have an assessment of the quality of the method, even though it is still on a sample. The precision seems fair on the first ten candidates, enough to be used in a semi-automatic way, coupled with a lexicographic analysis.

5 Related work

Among the methods proposed to collect synonymy information, two families can be distinguished according to the input they consider. Either a general dictionary is used (or more than one (Wu and Zhou, 2003)), or a corpus of unconstrained texts from which lexical distributions are computed (simple collocations or syntactic dependencies) (Lin, 1998; Freitag et al., 2005) . The approach of (Barzilay and McKeown, 2001) uses a related kind of resource: multiple translations of the same text, with additional constraints on availability, and problems of text alignment, for only a third of the results being synonyms (when compared to Wordnet).

A measure of similarity is almost always used to rank possible candidates. In the case of distributional approaches, similarity if determined from the appearance in similar contexts (Lin, 1998); in the case of dictionary-based methods, lexical relations are deduced from the links between words expressed in definitions of entries.

Approaches that rely on distributional data have two major drawbacks: they need a lot of data, generally syntactically parsed sentences, that is not always available for a given language (English is an exception), and they do not discriminate well among lexical relations (mainly hyponyms, antonyms, hypernyms) (Weeds et al., 2004) . Dictionary-based

approaches address the first problem since dictionaries are readily available for a lot of language, even electronically, and this is the *raison d'être* of our effort. As we have seen here, it is not an obvious task to sort related terms with respect to synonymy, hypernymy, etc, just as with distribution approaches.

A lot of work has been done to extract lexical relations from the definitions taken in isolation (mostly for ontology building), see recently (Nichols et al., 2005), with a syntactic/semantic parse, with usually results around 60% of precision (that can be compared with the same baseline we used, all words in the definition with the same category), on dictionaries with very small definitions (and thus a higher proportions of synonyms and hypernyms). Estimating the recall of such methods have not been done.

Using dictionaries as network of lexical items or senses has been quite popular for word sense disambiguation (Veronis and Ide, 1990; H.Kozima and Furugori, 1993; Niwa and Nitta, 1994) before losing ground to statistical approaches, even though (Gaume et al., 2004; Mihalcea et al., 2004) tried a revival of such methods. Both (Ho and Cédrick, 2004) and (Blondel et al., 2004) build a graph of lexical items from a dictionary in a manner similar to ours. In the first case, the method used to compute similarity between two concepts (or words) is restricted to neighbors, in the graph, of the two concepts; in the second case, only directly related words are considered as potential candidates for synonymy: for two words to be considered synonyms, one has to appear in the definition of another. In both cases, only 6 or 7 words have been used as a test of synonymy, with a validation provided by the authors with "related terms" (an unclear notion) considered correct. The similarity measure itself was evaluated on a set of related terms from (Miller and Charles, 1991), as in (Budanitsky and Hirst, 2001; Banerjee and Pedersen, 2003), with seemingly good results, but semantically related terms is a very different notion ("car" and "tire" for instance are semantically related terms, and thus considered similar).
We do not know of any dictionary-based graph approach which have been given a larger evaluation of its results. Parsing definitions in isolation prevents a complete coverage (we estimated that only 30% of

synonyms pairs in the TLF can be found from definitions).

As for distributional approaches, (Barzilay and McKeown, 2001) gets a very high precision (around 90%) on valid paraphrases as judged by humans, among which 35% are synonymy relations in Wordnet, 32% are hypernyms, 18% are coordinate terms. Discriminating among the paraphrases types is not addressed. Other approaches usually consider either given sets of synonyms among which one is to be chosen (for a translation for instance) (Edmonds and Hirst, 2002) or must choose a synonym word against unrelated terms in the context of a synonymy test (Freitag et al., 2005), a seemingly easier task than actually proposing synonyms. (Lin, 1998) proposes a different methodology for evaluation of candidate synonyms, by comparing similarity measures of the terms he provides with the similarity measures between them in Wordnet, using various semantic distances. This makes for very complex evaluation procedures without an intuitive interpretation, and there is no assessment of the quality of the automated thesaurus.

6 Conclusion

We have developed a general method to extract near-synonyms from a dictionary, improving on the two baselines. There is some arbitrariness in the parameters we used, but we believe the parameters are rather intuitive wrt to graph concepts.[7] There is room for improvement obviously, also for a combination with other methods to filter synonyms (with frequency estimates for instance, such as tf.idf or mutual information measures).

Clearly the advantage of using a dictionary is retained: there is no restriction of coverage, and we could have used a specialised dictionary to build a specialised thesaurus. We have provided an assessment of the quality of the results, although there is not much to compare it to (to the best of our knowledge), since previous accounts only had cursory evaluation.

---

[7]The lexical graph can be explored at http://prox.irit.fr.

References

S. Banerjee and T. Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of IJCAI-03*, Acapulco, Mexico.

Regina Barzilay and Kathleen R. McKeown. 2001. Extracting paraphrases from a parallel corpus. In *Proceedings of the 39th ACL*, pages 00–00, Toulouse.

Vincent D. Blondel, Anahì Gajardo, Maureen Heymans, Pierre Senellart, and Paul Van Dooren. 2004. A measure of similarity between graph vertices: Applications to synonym extraction and web searching. *SIAM Review*, 46(4):647–666.

A. Budanitsky and G. Hirst. 2001. Semantic distance in wordnet: An experimental, application-oriented evaluation of five measures. In *Workshop on WordNet and Other Lexical Resources, NAACL 2001*, Pittsburgh.

Philip Edmonds and Graeme Hirst. 2002. Near-Synonymy and lexical choice. *Computational Linguistics*, 28(2):105–144.

Dayne Freitag, Matthias Blume, John Byrnes, Edmond Chow, Sadik Kapadia, Richard Rohwer, and Zhiqiang Wang. 2005. New experiments in distributional representations of synonymy. In *Proceedings of CoNLL*, pages 25–32, Ann Arbor, Michigan, June. Association for Computational Linguistics.

B. Gaume, N. Hathout, and P. Muller. 2004. Word sense disambiguation using a dictionary for sense similarity measure. In *Proceedings of Coling 2004*, volume II, pages 1194–1200, Genève.

B. Gaume, F. Venant, and B. Victorri. 2005. Hierarchy in lexical organization of natural language. In D. Pumain, editor, *Hierarchy in natural and social sciences*, Methodos series, pages 121–143. Kluwer.

H.Kozima and T. Furugori. 1993. Similarity between words computed by spreading activation on an english dictionary. In *Proceedings of the EACL*, pages 232–239.

Ngoc-Diep Ho and Fairon Cédrick. 2004. Lexical similarity based on quantity of information exchanged - synonym extraction. In *Proc. of Intl. Conf. RIVF'04*, Hanoi.

Irene Langkilde and Kevin Knight. 1998. Generation that exploits corpus-based statistical knowledge. In *Proceedings of COLING-ACL '98*, volume 1, pages 704–710, Montreal.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of COLING-ACL '98*, volume 2, pages 768–774, Montreal.

Rada Mihalcea, Paul Tarau, and Elizabeth Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *Proceedings of Coling 2004*, Geneva.

GA Miller and WG Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.

Eric Nichols, Francis Bond, and Daniel Flickinger. 2005. Robust ontology acquisition from machine-readable dictionaries. In *Proceedings of IJCAI'05*.

Yoshiki Niwa and Yoshihiko Nitta. 1994. Co-occurrence vectors from corpora vs. distance vectors from dictionaries. In *Proceedings of Coling 1994*.

Sabine Ploux and Hyungsuk Ji. 2003. A model for matching semantic maps between languages (French/English, English/French). *Computational Linguistics*, 29(2):155–178.

J. Veronis and N.M. Ide. 1990. Word sense disambiguation with very large neural networks extracted from machine readable dictionaries. In *COLING-90: Proceedings of the 13th International Conference on Computational Linguistics*, volume 2, pages 389–394, Helsinki, Finland.

Julie Weeds, David Weir, and Diana McCarthy. 2004. Characterising measures of lexical distributional similarity. In *Proceedings of Coling 2004*, pages 1015–1021, Geneva, Switzerland, Aug 23–Aug 27. COLING.

Hua Wu and Ming Zhou. 2003. Optimizing synonyms extraction with mono and bilingual resources. In *Proceedings of the Second International Workshop on Paraphrasing*, Sapporo, Japan. Association for Computational Linguistics.

Ingrid Zukerman, Sarah George, and Yingying Wen. 2003. Lexical paraphrasing for document retrieval and node identification. In *Proceedings of the Second International Workshop on Paraphrasing*, Sapporo, Japan. Association for Computational Linguistics.