

# Effectiveness of Indirect Dependency for Automatic Synonym Acquisition

Masato HAGIWARA, Yasuhiro OGAWA, and Katsuhiko TOYAMA

Nagoya University, Furo-cho, Chikusa-ku, Nagoya, Japan  
{hagiwara,yasuhiro,toyama}@k1.i.is.nagoya-u.ac.jp

**Abstract.** Since synonyms are important lexical knowledge, various methods have been proposed for automatic synonym acquisition. Whereas most of the methods are based on the *distributional hypothesis* and utilize contextual clues, little attention has been paid to what kind of contextual information is useful for the purpose. As one of the ways to augment contextual information, we propose the use of *indirect dependency*, i.e. relation between two words related via two contiguous dependency relations. The evaluation has shown that the performance improvement over normal direct dependency is dramatic, yielding comparable results with surrounding words as context, even with smaller co-occurrence data.

## 1 Introduction

Lexical knowledge is one of the most fundamental but important resources for natural language processing. Among various kinds of lexical relations, synonyms are used in a broad range of applications such as query expansion for information retrieval [8] and automatic thesaurus construction [9].

Various methods [7, 10] have been proposed for automatic synonym acquisition. They are often based on the distributional hypothesis [6], which states that semantically similar words share similar contexts, and they can be roughly viewed as the combinations of these two steps: context extraction and similarity calculation. The former extracts useful information such as dependency relations of words from corpora. The latter calculates how semantically similar two given words are, based on the co-occurrence counts or frequency distributions acquired in the first step, using similarity models such as mutual information.

However, whereas many methods employ the context-based similarity calculation, almost no attention has been paid to what kind of contextual information is useful for word featuring in terms of synonym acquisition.

For example, Ruge [13] proposed the use of dependency structure of sentences to detect term similarities for automatic thesaurus construction and showed the evaluation result to be encouraging, but neither the further investigation of dependency selection nor the comparison with other kinds of contextual information is provided. Lin [10] used a broad-coverage parser to extract wider range of grammatical relationship and showed the possibility that other kind of dependency relations in addition to subject and object was contributing, although it is still not clear what kind of relations affects the performance, or to what extent.

Few exceptions include Curran’s [3], where they compared context extractors such as window extractor and shallow- and deep-parsing extractor. Their observation, however, doesn’t accompany discussion concerning the qualitative difference of the context extractors and its causes. Because the choice of useful contextual information has a critical importance on the performance, further investigations on which types of contexts are essentially contributing are required.

As one of the ways to augment the contextual information, this paper proposes the use of *indirect dependency*, and shows its effectiveness for automatic synonym acquisition. We firstly extract *direct dependency* using RASP parser [1] from three different corpora, then extend it to indirect dependency which includes the relations composed from two or more contiguous dependency relations. The contexts corresponding direct and indirect dependency are extracted, and co-occurrences of words and their contexts are obtained. Because the details of similarity calculation is not the scope of this paper, widely used vector space model, tf.idf weighting, and cosine measure are adopted. The acquisition performance is evaluated using two automatic evaluation measures: average precision (AP) and correlation coefficient (CC) based on three existing thesauri.

This paper is organized as follows: in Section 2 we mention the preliminary experiment result of contextual information selection, along with the background of how we get to choose the indirect dependency. Sections 3 and 4 detail the formalization and the context extraction for indirect dependency. Section 5 briefly describes the synonym acquisition model we used, and in the following Section 6 the evaluation method is detailed. Section 7 provides the experimental conditions and results, followed by Section 8 which concludes this paper.

## 2 Context Selection

In this section, we show the result of the preliminary experiment of contextual information selection, and describe how we came up with the idea that the extension of normal direct dependency could be beneficial. Here we focused on the following three kinds of contextual information for comparison:

- **dep**: direct dependency; contexts extracted from the grammatical relations computed by RASP parser.
- **prox**: word proximity; surrounding words, i.e. words which locate within the window centered at a target word, and their relative positions. For example, a context having “the” on the left is represented as `L1:the`. We set the window radius to 3 in this paper.
- **sent**: sentence co-occurrence; sentence id in which the words occur. The underlying assumption of using this information is that words which occur in the same sentence are likely to share similar topics.

The overall experimental framework and evaluation scheme are same as the ones mentioned in the later sections. AP is the precision of acquired synonyms and CC is how similar the obtained similarity is correlated with WordNet’s. The result, shown in Figure 1, suggests the superiority of **prox** over **dep** although the window range to capture the surrounding words is rather limited. This result

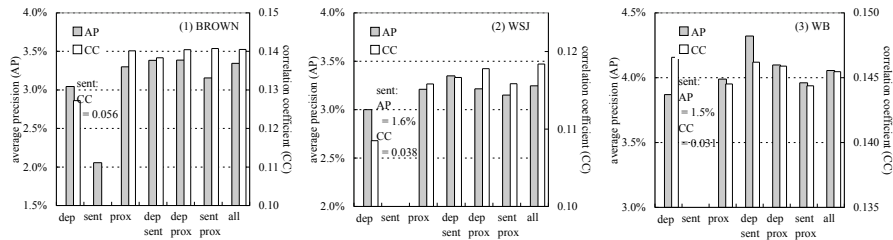


Fig. 1. Contextual information selection performances

makes us wonder what types of contextual information other than dependency are contained in the “difference” of two sets, and we suspect this remainder causes the significant improvement on the performance. In other words, there should be some useful contextual information contained in **prox** but not in **dep**.

We notice here that the word relations in **dep** are limited only to two words which have *direct* dependency between them, but there may be some words within the proximity window that indirectly have relations not captured by **dep**, e.g. a subject and an object sharing the same verb in a sentence. To capture this, we utilize this indirect dependency, which is detailed in the following section.

### 3 Indirect Dependency

This section describes the formalization of indirect dependency we adopted. Here we consider the dependency relations in a certain sentence  $s$  as a binary relation  $D$  over  $W = \{w_1, \dots, w_n\}$  i.e.  $D \subset W \times W$ , where  $w_1, \dots, w_n$  are the words in  $s$ . Since no words can be dependent or modifier of itself,  $D$  is irreflexive.

We define the composition of dependency  $D^2 = D \circ D$  as *indirect dependency* where two words are related via two dependency relation edges. Each edge has labels assigned such as **subj** and **dobj** which specify what kind of syntactic relations the head and modifier possess. When an indirectly related pair  $r_i \in D^2$  is composed from  $r_j \in D$  with a label  $l_j$  and  $r_k \in D$  with a label  $l_k$ , the label of  $r_i$  is also composed from  $l_j$  and  $l_k$ . We also define multiple composition of dependency recursively:  $D^1 = D, \forall n > 1. D^n = D^{n-1} \circ D$ . These are also indirect dependency relations in a broad sense. Notice here that  $D^n$  ( $n > 1$ ) can generally include reflexive relations, but it is clear that such relations don’t serve as useful word features, so we re-define the composition operation so that the composed relation doesn’t include any reflexive edges, i.e.  $D \circ D - \{(w, w) | w \in W\}$ .

### 4 Context Extraction

This section describes how to extract the contexts corresponding to direct and indirect dependency relations. First, the direct dependency is computed for each sentence, then the corresponding direct and indirect contexts are constructed from the dependency. As the extraction of comprehensive grammatical relations is a difficult task, RASP Toolkit was utilized to extract this kind of word relations. RASP analyzes sentences and extracts the dependency structure called grammatical relations (GRs). Take the following sentence for example:

---

```

(ncsubj be Shipment _)
(aux be have)
(xcomp _ be level)
(ncmod _ be relatively)
(ccomp _ level note)
(ncmod _ note since)
(ncsubj note Department _)
(det Department the)
(ncmod _ Department Commerce)
(dobj since January)

```

---

**Fig. 2.** Examples of extracted GRs

---

```

Shipment - (ncsubj be * _)
have - (aux be *)
be - (ncsubj * Shipment _)
be - (aux * have)
be - (xcomp _ * level)
be - (ncmod _ * relatively)
relatively - (ncmod _ be *)
:
since - (ncmod _ note *)
January - (dobj since *)
:

```

---

**Fig. 3.** Examples of contexts.

Shipments have been relatively level since January, the Commerce Department noted.

RASP extract GRs as n-ary relations as shown in Figure 2. While the RASP outputs are n-ary relations in general, what we need here is pairs of words and contexts, so we extract co-occurrences of words and direct contexts  $C^1$  corresponding to  $D^1$ , by extracting the target word from the relation and replacing the slot by an asterisk “\*”, as shown in Figure 3. This operation corresponds to creating word-context pairs by converting a pair  $r \in D^1$  of a head  $h$  and a dependent  $d$  with a label  $l_i$  into the pair  $(h, l_i:d)$ . If  $(h, l_i:d) \in C^1$ , then  $(d, l_j:h) \in C^1$  also holds, where the label  $l_j$  is the *inverse* of  $l_i$ , as the two pairs **have - (aux be \*)** and **be - (aux \* have)** show in the figure. We treated all the slots except for head and modifier as the extra information and included them as the labels.

The co-occurrence of words and indirect contexts,  $C^2$ , which corresponds to indirect dependency  $D^2$  is generated from  $C^1$ . For example,  $D^2$  contains the indirect relation **Shipment - be - level** composed from **(ncsubj be Shipment \_)** and **(xcomp \_ be level)**. The context of **Shipment** extracted from this indirect relation is then formed by embedding the context of **be: (xcomp \_ \* level)** into the slot **be** of the context of **Shipment: (ncsubj be \* \_)**, which yields **Shipment - (ncsubj (xcomp \_ \* level) \* \_)**. Similarly, the indirect relation “January is the direct object of since, which in turn is modifying the verb note” is expressed as: **January - (dobj (ncmod \_ note \*) \*)**.

Co-occurrences of indirect contexts  $C^n$  ( $n \geq 3$ ) corresponding to the multiple composition  $D^n$  are derived analogously.  $C^3$ , for example, is yielded just by embedding  $C^1$  contexts into  $C^2$  contexts shown in the previous example.

## 5 Synonym Acquisition Method

The purpose of the current study is to investigate the effectiveness of indirect dependency relations, not the language or acquisition model itself, we simply employed one of the most commonly used method: vector space model (VSM)

and tf.idf weighting scheme, although they might not be the best choice according to the past studies. In this framework, each word  $w_i$  is represented as a vector  $\mathbf{w}_i$  whose elements are given by tf.idf, i.e. co-occurrence frequencies of words and contexts, weighted by normalized idf. That is, letting the number of distinct words and contexts in the corpus be  $N$  and  $M$ , co-occurrence frequency of word  $w_i$  and context  $c_j$  be  $\text{tf}(w_i, c_j)$ ,

$$\mathbf{w}_i = {}^t[\text{tf}(w_i, c_1) \cdot \text{idf}(c_1) \dots \text{tf}(w_i, c_M) \cdot \text{idf}(c_M)], \quad (1)$$

$$\text{idf}(c_j) = \frac{\log(N/\text{df}(c_j))}{\max_k \log(N/\text{df}(c_k))}, \quad (2)$$

where  $\text{df}(c_j)$  is the number of distinct words that co-occur with context  $c_j$ . The similarity between two words are then calculated using cosine of two vectors.

## 6 Evaluation

This section describes the two evaluation methods we employed — average precision (AP) and correlation coefficient (CC).

### 6.1 Average Precision

The first evaluation measure, average precision (AP), is a common evaluation scheme for information retrieval, which evaluates how accurately the methods are able to extract synonyms. We first prepare a set of *query words*, for which synonyms are obtained to evaluate the precision. We adopted the Longman Defining Vocabulary (LDV) <sup>1</sup> as the candidate set of query words. For each query word in LDV, three existing thesauri are consulted: Roget’s Thesaurus [4], Collins COBUILD Thesaurus [2], and WordNet. The union of synonyms obtained when the query word is looked up as a noun is used as the reference set, except for words marked as “idiom,” “informal,” “slang” and phrases comprised of two or more words. The query words for which no noun synonyms are found in any of the reference thesauri are omitted. For each of the remaining query words, the number of which turned out to be 771, the eleven precision values at 0%, 10%, ..., and 100% recall levels are averaged to calculate the final AP value.

### 6.2 Correlation Coefficient

The second evaluation measure is correlation coefficient (CC) between the target similarity and the *reference similarity*, i.e. the answer value of similarity for word pairs. The reference similarity is calculated based on the closeness of two words in the tree structure of WordNet. More specifically, the similarity between word  $w$  with senses  $w_1, \dots, w_{m_1}$  and word  $v$  with senses  $v_1, \dots, v_{m_2}$  is obtained as follows. Let the depth of node  $w_i$  and  $v_j$  be  $d_i$  and  $d_j$ , and the maximum depth of the common ancestors of both nodes be  $d_{\text{dca}}$ . The similarity is then

$$\text{sim}(w, v) = \max_{i,j} \text{sim}(w_i, v_j) = \max_{i,j} \frac{2 \cdot d_{\text{dca}}}{d_i + d_j}, \quad (3)$$

<sup>1</sup> [http://www.cs.utexas.edu/users/kbarker/working\\_notes/ldoce-vocab.html](http://www.cs.utexas.edu/users/kbarker/working_notes/ldoce-vocab.html).

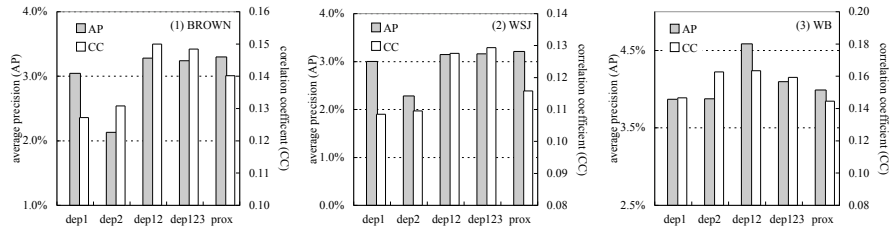


Fig. 4. Performance of the direct and indirect dependency relations

which takes the value between 0.0 and 1.0. Then, the value of CC is calculated as the correlation coefficient of reference similarities  $\mathbf{r} = (r_1, r_2, \dots, r_n)$  and target similarities  $\mathbf{s} = (s_1, s_2, \dots, s_n)$  over the word pairs in sample set  $P_s$ , which is created by choosing the most similar 2,000 word pairs from 4,000 random pairs. Every CC value in this paper is the average of 10 executions using 10 randomly created test sets to avoid the test-set dependency.

## 7 Experiments

Now we describe the evaluation results for indirect dependency.

### 7.1 Condition

We extracted contextual information from these three corpora: (1) Wall Street Journal (WSJ) (approx. 68,000 sentences, 1.4 million tokens), (2) Brown Corpus (BROWN) (approx. 60,000 sentences, 1.3 million tokens), both of which are contained in Treebank 3 [11], and (3) written sentences in WordBank (WB) (approx. 190,000 sentences, 3.5 million words) [2]. No additional annotation such as POS tags provided for Treebank was used. As shown in Sections 2 and 3, only relations (positions for `prox`) and word stems were used as context.

Since our purpose here is the automatic extraction of synonymous nouns, only the contexts for nouns are extracted. To distinguish nouns, using POS tags annotated by RASP, any words with POS tags APP, ND, NN, NP, PN, PP were labeled as nouns. We set a threshold  $t_f$  on occurrence frequency to filter out any words or contexts with low frequency and to reduce computational cost. More specifically, any words  $w$  such that  $\sum_c \text{tf}(w, c) < t_f$  and any contexts  $c$  such that  $\sum_w \text{tf}(w, c) < t_f$  were removed from the co-occurrence data.  $t_f$  was set to  $t_f = 5$  for WSJ and BROWN, and  $t_f = 15$  for WB.

### 7.2 Performance of Indirect Dependency

In this section, we experimented to confirm the effectiveness of indirect dependency. The performances of the following categories and combinations are evaluated: `prox`,  $C^1$  (`dep1`),  $C^2$  (`dep2`),  $C^1 \cup C^2$  (`dep12`), and  $C^1 \cup C^2 \cup C^3$  (`dep123`).

The evaluation result for three corpora is shown in Figure 4. We observe that whereas `prox` was better than the direct dependency `dep1` as shown in Section 2, the performance of the combination of direct and indirect dependency `dep12`

**Table 1.** Examples of acquired synonyms and their similarity for word “legislation”.

dep1		dep12	
word	similarity	word	similarity
law	0.328	law	0.280
circumstance	0.242	money	0.229
auspices	0.239	plan	0.227
rule	0.225	issue	0.227
supervision	0.227	rule	0.225
pressure	0.226	change	0.222
condition	0.224	system	0.218
control	0.225	project	0.216
microscope	0.218	company	0.214
violence	0.209	power	0.212

was comparable to or even better than **prox**, and the improvement over **dep1** was dramatic. Table 1 shows the examples of extracted synonyms. It is seen that using **dep12** improves the result, and instead of less relevant words such as “microscope” and “violence”, more relevant words like “plan” and “system” come up as the ten most similar words. Adding  $C^3$  to **dep12**, on the other hand, didn’t further improve the result, from which we can conclude that extending and augmenting  $C^1$  just one step is sufficient in practice.

As for the data size, the numbers of distinct co-occurrences of **prox** and **dep12** extracted from BROWN corpus were 899,385 and 686,782, respectively. These numbers are rough approximations of the computational costs of calculating similarities, which means that **dep12** is a good-quality context because it achieves better performance with smaller co-occurrence data than **prox**. On the other hand, the numbers of distinct contexts of **prox** and **dep12** were 10,624 and 30,985, suggesting that the more diverse the contexts are, the better the performance is likely to be. This result was observed for other corpora as well, and is consistent with the one that we have previously shown [5], that is, what is essential to the performance is not the quality or the quantity of the context, but its diversity.

It is thus concluded that we can attribute the superiority of **dep12** to its potential to greatly increase the contextual information variety, and although the extraction of dependency is itself a costly task, adding the extra **dep2** is a very reasonable augmentation which requires little extra computational cost, aside from the marginal increase of the resultant co-occurrence data.

## 8 Conclusion

In this study, we proposed the use of indirect dependency composed from direct dependency to enhance the contextual information for automatic synonym acquisition. The indirect contexts were constructed from the direct dependency extracted from three corpora, and the acquisition result was evaluated based on two evaluation measures, AP and CC using the existing reference thesauri.

We showed that the performance improvement of indirect dependency over the direct dependency was dramatic. Also, the indirect contexts showed better

results when compared to surrounding words even with smaller co-occurrence data, which means that the indirect context is effective in terms of quality as well as computational cost. The use of indirect dependency is an very efficient way to increase the context variety, taking into consideration the fact that the diversity of contexts is likely to be essential to the acquisition performance.

Because we started from the “difference” of dependency relations and word proximity, the investigation of other kinds of useful contextual information should be conducted in the future. There are also some studies including Pado’s [12] that make the most of dependency paths in the sentence, but their model does not take into account the dependency label. This increases the granularity of contexts and its effect is an open issue which we should bring up in another article. The application to other categories of words or the extraction of semantic relations other than synonyms is the future work.

## References

1. Briscoe, T., Carroll, J., Watson, R.: The Second Release of the RASP System. Proc. COLING/ACL 2006 Interactive Presentation Sessions (2006) 77–80.
2. Collins.: Collins COBUILD Major New Edition CD-ROM. HyperCollins (2002).
3. Curran, James R., Moens, M.: Improvements in Automatic Thesaurus Extraction. Proc. SIGLEX (2002) 59–66.
4. Editors of the American Heritage Dictionary: Roget’s II: The New Thesaurus, 3rd ed. Boston: Houghton Mifflin (1995).
5. Hagiwara, M., Ogawa, Y., Toyama, K.: Selection of Effective Contextual Information for Automatic Synonym Acquisition. Proc. COLING/ACL (2006) 353–360.
6. Harris, Z.: Distributional Structure. Katz, J. J. (ed.): The Philosophy of Linguistics, Oxford University Press (1985) 26–47.
7. Hindle, D.: Noun classification from predicate-argument structures. Proc. ACL (1990) 268–275.
8. Jing, Y., Croft, B.: An association thesaurus for information retrieval. Proc. RIAO (1994) 146–160.
9. Kojima, H., Ito, A.: Adaptive Scaling of a Semantic Space. IPSJ SIGNotes Natural Language, NL108-13, (1995) 81–88. (in Japanese)
10. Lin, D.: Automatic retrieval and clustering of similar words. Proc. COLING/ACL (1998) 786–774.
11. Marcus, M. P., Santorini, B., Marcinkiewicz, M. A.: Building a large annotated corpus of English: The Penn treebank. Computational Linguistics, 19(2) (1994) 313–330.
12. Pado, S., Lapata, M.: Constructing semantic space models from parsed corpora. Proc. ACL (2003) 128–135.
13. Ruge, G.: Automatic detection of thesaurus relations for information retrieval applications. Foundations of Computer Science: Potential - Theory - Cognition, LNCS, vol. 1337 (1997) 499–506.