A Supervised Learning Approach to Automatic Synonym Identification based on Distributional Features

Masato Hagiwara
Graduate School of Information Science
Nagoya University
Furo-cho, Chikusa-ku, Nagoya 464-8603, JAPAN
hagiwara@kl.i.is.nagoya-u.ac.jp

Abstract

Distributional similarity has been widely used to capture the semantic relatedness of words in many NLP tasks. However, various parameters such as similarity measures must be hand-tuned to make it work effectively. Instead, we propose a novel approach to synonym identification based on supervised learning and *distributional features*, which correspond to the commonality of individual context types shared by word pairs. Considering the integration with *pattern-based features*, we have built and compared five synonym classifiers. The evaluation experiment has shown a dramatic performance increase of over 120% on the F-1 measure basis, compared to the conventional similarity-based classification. On the other hand, the pattern-based features have appeared almost redundant.

1 Introduction

Semantic similarity of words is one of the most important lexical knowledge for NLP tasks including word sense disambiguation and automatic thesaurus construction. To measure the semantic relatedness of words, a concept called *distributional similarity* has been widely used. Distributional similarity represents the relatedness of two words by the commonality of contexts the words share, based on the *distributional hypothesis* (Harris, 1985), which states that semantically similar words share similar contexts.

A number of researches which utilized distributional similarity have been conducted, including (Hindle, 1990; Lin, 1998; Geffet and Dagan, 2004) and many others. Although they have been successful in acquiring related words, various parameters such as similarity measures and weighting are involved. As Weeds et al. (2004) pointed out, "it is not at all obvious that one universally best measure exists for all application," thus they must be tuned by hand in an ad-hoc manner. The fact that no theoretic basis is given is making the matter more difficult.

On the other hand, if we pay attention to lexical knowledge acquisition in general, a variety of systems which utilized *syntactic patterns* are found in the literature. In her landmark paper in the field, Hearst (1992) utilized syntactic patterns such as "such X as Y" and "Y and other X," and extracted hypernym/hyponym relation of X and Y. Roark and Charniak (1998) applied this idea to extraction of words which belong to the same categories, utilizing syntactic relations such as conjunctions and appositives. What is worth attention here is that supervised machine learning is easily incorporated with syntactic patterns. For example, Snow et al. (2004) further extended Hearst's idea and built hypernym classifiers based on machine learning and syntactic pattern-based features, with a considerable success.

These two independent approaches, distributional similarity and syntactic patterns, were finally integrated by Mirkin et al. (2006). Although they reported that their system successfully improved the performance, it did not achieve a complete integration and was still relying on an independent module to compute the similarity. This configuration inherits a large portion of drawbacks of the similarity-based approach mentioned above. To achieve a full integration of both approaches, we suppose that re-

Proceedings of the ACL-08: HLT Student Research Workshop (Companion Volume), pages 1–6,
Columbus, June 2008. ©2008 Association for Computational Linguistics

formalization of similarity-based approach would be essential, as pattern-based approach is enhanced with the supervised machine learning.

In this paper, we propose a novel approach to automatic synonym identification based on supervised learning technique. Firstly, we re-formalize synonym acquisition as a classification problem: one which classifies *word pairs* into synonym/non-synonym classes, without depending on a single value of distributional similarity. Instead, classification is done using a set of *distributional features*, which correspond to the degree of commonality of individual context types shared by word pairs. This formalization also enables to incorporate pattern-based features, and we finally build five classifiers based on distributional and/or pattern-based features. In the experiment, their performances are compared in terms of synonym acquisition precision and recall, and the differences of actually acquired synonyms are to be clarified.

The rest of this paper is organized as follows: in Sections 2 and 3, distributional and pattern-based features are defined, along with the extraction methods. Using the features, in Section 4 we build five types of synonym classifiers, and compare their performances in Section 5. Section 6 concludes this paper, mentioning the future direction of this study.

2 Distributional Features

In this section, we firstly describe how we extract contexts from corpora and then how distributional features are constructed for word pairs.

2.1 Context Extraction

We adopted dependency structure as the context of words since it is the most widely used and well-performing contextual information in the past studies (Ruge, 1997; Lin, 1998). In this paper the sophisticated parser RASP Toolkit 2 (Briscoe et al., 2006) was utilized to extract this kind of word relations. We use the following example for illustration purposes: *The library has a large collection of classic books by such authors as Herrick and Shakespeare.* RASP outputs the extracted dependency structure as n-ary relations as follows:

```
(ncsubj have library _)
(dobj have collection)
(det collection a)
(ncmod _ collection large)
(iobj collection of)
(dobj of book)
(ncmod _ book by)
(dobj by author)
(det author such)
(ncmod _ author as)
... ,
```

whose graphical representation is shown in Figure 1.

While the RASP outputs are n-ary relations in general, what we need here is co-occurrences of words and contexts, so we extract the set of co-occurrences of stemmed words and contexts by taking out the target word from the relation and replacing the slot by an asterisk “*”:

```
library     - (ncsubj have * _)
library     - (det * The)
collection  - (dobj have *)
collection  - (det * a)
collection  - (ncmod _ * large)
collection  - (iobj * of)
book        - (dobj of *)
book        - (ncmod _ * by)
book        - (ncmod _ * classic)
author      - (dobj by *)
author      - (det * such)
...
```

Summing all these up produces the raw co-occurrence count $N(w, c)$ of the word w and the context c. In the following, the set of context types co-occurring with the word w is denoted as $C(w)$, i.e., $C(w) = \{c|N(w, c) > 0\}$.

2.2 Feature Construction

Using the co-occurrences extracted above, we define distributional features $f_j^D(w_1, w_2)$ for the word pair (w_1, w_2). The feature value f_j^D is determined so that it represents the degree of commonality of the context c_j shared by the word pair. We adopted *pointwise total correlation*, one of the generalizations of pointwise mutual information, as the feature value:

$$f_j^D(w_1, w_2) = \log \frac{P(w_1, w_2, c_j)}{P(w_1)P(w_2)P(c_j)}. \quad (1)$$

The advantage of this feature construction is that, given the independence assumption between the words w_1 and w_2, the feature value is easily calculated as the simple sum of two corresponding pointwise mutual information weights as:

$$f_j^D(w_1, w_2) = \mathrm{PMI}(w_1, c_j) + \mathrm{PMI}(w_2, c_j), \quad (2)$$

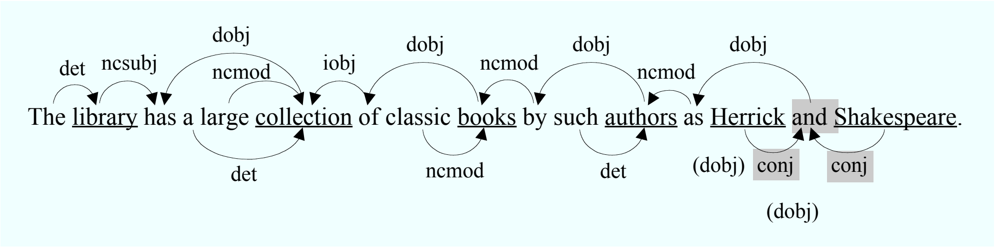
dobj
det
ncsubj
ncmod
iobj
dobj
ncmod
dobj
ncmod
dobj
The library has a large collection of classic books by such authors as Herrick and Shakespeare.
det
ncmod
det
(dobj)
conj
conj
(dobj)

Figure 1: Dependency structure of the example sentence, along with conjunction shortcuts (dotted lines).

where the value of PMI, which is also the weights $\mathrm{wgt}(w_i, c_j)$ assigned for distributional similarity, is calculated as:

$$\mathrm{wgt}(w_i, c_j) = \mathrm{PMI}(w_i, c_j) = \log \frac{P(w_i, c_j)}{P(w_i)P(c_j)}. \quad (3)$$

There are three things to note here: when $N(w_i, c_j) = 0$ and PMI cannot be defined, then we define $\mathrm{wgt}(w_i, c_j) = 0$. Also, because it has been shown (Curran and Moens, 2002) that negative PMI values worsen the distributional similarity performance, we bound PMI so that $\mathrm{wgt}(w_i, c_j) = 0$ if $\mathrm{PMI}(w_i, c_j) < 0$. Finally, the feature value $f_j^D(w_1, w_2)$ is defined as shown in Equation (2) only when the context c_j co-occurs with both w_1 and w_2. In other words, $f_j^D(w_1, w_2) = 0$ if $\mathrm{PMI}(w_1, c_j) = 0$ and/or $\mathrm{PMI}(w_2, c_j) = 0$.

3 Pattern-based Features

This section describes the other type of features, extracted from syntactic patterns in sentences.

3.1 Syntactic Pattern Extraction

We define *syntactic patterns* based on dependency structure of sentences. Following Snow et al. (2004)'s definition, the syntactic pattern of words w_1, w_2 is defined as the concatenation of the words and relations which are on the dependency path from w_1 to w_2, not including w_1 and w_2 themselves.

The syntactic pattern of word *authors* and *books* in Figure 1 is, for example, `dobj:by:ncmod`, while that of *authors* and *Herrick* is `ncmod-of:as:dobj-of:and:conj-of`. Notice that, although not shown in the figure, every relation has a reverse edge as its counterpart, with the direction opposite and the postfix "`-of`" attached to the label. This allows to follow the relations in reverse, increasing the flexibility and expressive power of patterns.

In the experiment, we limited the maximum length of syntactic path to five, meaning that word pairs having six or more relations in between were disregarded. Also, we considered *conjunction shortcuts* to capture the lexical relations more precisely, following Snow et al. (2004). This modification cuts short the `conj` edges when nouns are connected by conjunctions such as *and* and *or*. After this shortcut, the syntactic pattern between *authors* and *Herrick* is `ncmod-of:as:dobj-of`, and that of *Herrick* and *Shakespeare* is `conj-and`, which is a newly introduced special symmetric relation, indicating that the nouns are mutually conjunctional.

3.2 Feature Construction

After the corpus is analyzed and patterns are extracted, the pattern based feature $f_k^P(w_1, w_2)$, which corresponds to the syntactic pattern p_k, is defined as the conditional probability of observing p_k given that the pair (w_1, w_2) is observed. This definition is similar to (Mirkin et al., 2006) and is calculated as:

$$f_k^P(w_1, w_2) = P(p_k | w_1, w_2) = \frac{N(w_1, w_2, p_k)}{N(w_1, w_2)}. \quad (4)$$

4 Synonym Classifiers

Now that we have all the features to consider, we construct the following five classifiers. This section gives the construction detail of the classifiers and corresponding feature vectors.

Distributional Similarity (DSIM) DSIM classifier is simple acquisition relying only on distributional similarity, not on supervised learning. Similar to conventional methods, distributional similarity between words w_1 and w_2, $\mathrm{sim}(w_1, w_2)$, is calculated for each word pair using Jaccard coefficient:

$$\frac{\sum_{c \in C(w_1) \cap C(w_2)} \min(\mathrm{wgt}(w_1, c), \mathrm{wgt}(w_2, c))}{\sum_{c \in C(w_1) \cup C(w_2)} \max(\mathrm{wgt}(w_1, c), \mathrm{wgt}(w_2, c))},$$

considering the preliminary experimental result. A threshold is set on the similarity and classification is performed based on whether the similarity is above or below of the given threshold. How to optimally set this threshold is described later in Section 5.1.

Distributional Features (DFEAT) DFEAT classifier does not rely on the conventional distributional similarity and instead uses the distributional features described in Section 2. The feature vector $\vec{v}$ of a word pair (w_1, w_2) is constructed as:

$$\vec{v} = (f_1^D,\ ...,\ f_M^D). \tag{5}$$

Pattern-based Features (PAT) This classifier PAT uses only pattern-based features, essentially the same as the classifier of Snow et al. (2004). The feature vector is:

$$\vec{v} = (f_1^P,\ ...,\ f_K^P). \tag{6}$$

Distributional Similarity and Pattern-based Features (DSIM-PAT) DSIM-PAT uses the distributional similarity of pairs as a feature, in addition to pattern-based features. This classifier is essentially the same as the integration method proposed by Mirkin et al. (2006). Letting $f^S = \mathrm{sim}(w_1, w_2)$, the feature vector is:

$$\vec{v} = (f^S,\ f_1^P,\ ...,\ f_K^P). \tag{7}$$

Distributional and Pattern-based Features (DFEAT-PAT) The last classifier, DFEAT-PAT, truly integrates both distributional and pattern-based features. The feature vector is constructed by replacing the f^S component of DSIM-PAT with distributional features $f_1^D,\ ...,\ f_M^D$ as:

$$\vec{v} = (f_1^D,\ ...,\ f_M^D,\ f_1^P,\ ...,\ f_K^P). \tag{8}$$

5 Experiments

Finally, this section describes the experimental setting and the comparison of synonym classifiers.

5.1 Experimental Settings

Corpus and Preprocessing As for the corpus, New York Times section (1994) of English Gigaword [1], consisting of approx. 46,000 documents, 922,000 sentences, and 30 million words, was analyzed to obtain word-context co-occurrences.

This can yield 10,000 or more context types, thus we applied feature selection and reduced the dimensionality. Firstly, we simply applied frequency cut-off to filter out any words and contexts with low frequency. More specifically, any words w such that $\sum_c N(w, c) < \theta_f$ and any contexts c such that $\sum_w N(w, c) < \theta_f$, with $\theta_f = 5$, were removed. DF (document frequency) thresholding is then applied, and context types with the lowest values of DF were removed until 10% of the original contexts were left. We verified through a preliminary experiment that this feature selection keeps the performance loss at minimum. As a result, this process left a total of 8,558 context types, or feature dimensionality.

The feature selection was also applied to pattern-based features to avoid high sparseness — only syntactic patterns which occurred more than or equal to 7 times were used. The number of syntactic pattern types left after this process is 17,964.

Supervised Learning Training and test sets were created as follows: firstly, the nouns listed in the Longman Defining Vocabulary (LDV) [2] were chosen as the target words of classification. Then, all the LDV pairs which co-occur more than or equal to 3 times with any of the syntactic patterns, i.e., $\{(w_1, w_2) | w_1, w_2 \in \mathrm{LDV}, \sum_p N(w_1, w_2, p) \geq 3\}$ were classified into synonym/non-synonym classes as mentioned in Section 5.2. All the positive-marked pair, as well as randomly chosen 1 out of 5 negative-marked pairs, were collected as the *example set* E. This random selection is to avoid extreme bias toward the negative examples. The example set E ended up with 2,148 positive and 13,855 negative examples, with their ratio being approx. 6.45.

The example set E was then divided into five partitions to conduct five-fold cross validation, of which four partitions were used for learning and the one for testing. SVM^{light} was adopted for machine learning, and RBF as the kernel. The parameters, i.e., the similarity threshold of DSIM classifier, gamma parameter of RBF kernel, and the cost-factor j of SVM, i.e., the ratio by which training errors on positive examples outweight errors on negative ones,

[1] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

[2] http://www.cs.utexas.edu/users/kbarker/working_notes/ldoce-vocab.html

Table 1: Performance comparison of synonym classifiers

Classifier	Precision	Recall	F-1
DSIM	33.13%	49.71%	39.76%
DFEAT	95.25%	82.31%	88.30%
PAT	23.86%	45.17%	31.22%
DSIM-PAT	30.62%	51.34%	38.36%
DFEAT-PAT	95.37%	82.31%	88.36%

were optimized using one of the 5-fold cross validation train-test pair on the basis of F-1 measure. The performance was evaluated for the other four train-test pairs and the average values were recorded.

5.2 Evaluation

To test whether or not a given word pair (w_1, w_2) is a synonym pair, three existing thesauri were consulted: Roget's Thesaurus (Roget, 1995), Collins COBUILD Thesaurus (Collins, 2002), and WordNet (Fellbaum, 1998). The union of synonyms obtained when the head word is looked up as a noun is used as the answer set, except for words marked as "idiom," "informal," "slang" and phrases comprised of two or more words. The pair (w_1, w_2) is marked as synonyms if and only if w_2 is contained in the answer set of w_1, or w_1 is contained in that of w_2.

5.3 Classifier Performance

The performances, i.e., precision, recall, and F-1 measure, of the five classifiers were evaluated and shown in Table 1. First of all, we observed a drastic improvement of DFEAT over DSIM — over 120% increase of F-1 measure. When combined with pattern-based features, DSIM-PAT showed a slight recall increase compared to DSIM, partially reconfirming the favorable integration result of (Mirkin et al., 2006). However, the combination DFEAT-PAT showed little change, meaning that the discriminative ability of DFEAT was so high that pattern-based features were almost redundant. To note, the performance of PAT was the lowest, reflecting the fact that synonym pairs rarely occur in the same sentence, making the identification using only syntactic pattern clues even more difficult.

The reason of the drastic improvement is that, as far as we speculate, the supervised learning may have favorably worked to cause the same effect as automatic feature selection technique. Features with high discriminative power may have been automatically promoted. In the distributional similarity setting, in contrast, the contributions of context types are uniformly fixed. In order to elucidate what is happening in this situation, the investigations on machine learning settings, as well as algorithms other than SVM should be conducted as the future work.

5.4 Acquired Synonyms

In the second part of this experiment, we further investigated what kind of synonyms were actually acquired by the classifiers. The targets are not LDV, but all of 27,501 unique nouns appeared in the corpus, because we cannot rule out the possibility that the high performance seen in the previous experiment was simply due to the rather limited target word settings. The rest of the experimental setting was almost the same as the previous one, except that the construction of training set is rather artificial — we used all of the 18,102 positive LDV pairs and randomly chosen 20,000 negative LDV pairs.

Table 2 lists the acquired synonyms of *video* and *program*. The results of DSIM and DFEAT are ordered by distributional similarity and the value of decision function of SVM, respectively. Notice that because neither word is included in LDV, all the pairs of the query and the words listed in the table are guaranteed to be excluded from the training set.

The result shows the superiority of DFEAT over DSIM. The irrelevant words (marked by "*" by human judgement) seen in the DSIM list are demoted and replaced with more relevant words in the DFEAT list. We observed the same trend for lower ranked words and other query words.

6 Conclusion and Future Direction

In this paper, we proposed a novel approach to automatic synonym identification based on supervised machine learning and distributional features. For this purpose, we re-formalized synonym acquisition as a classification problem, and constructed the features as the total correlation of pairs and contexts. Since this formalization allows to integrate pattern-based features in a seamless way, we built five classifiers based on distributional and/or pattern-based features. The result was promising, achieving more than 120% increase over conventional DSIM classi-

Table 2: Acquired synonyms of *video* and *program*

For query word: *video*

Rank	DSIM	DFEAT
1	computer	computer
2	television	television
3	movie	multimedia
4	film	communication
5	food*	entertainment
6	multimedia	advertisement
7	drug*	food*
8	entertainment	recording
9	music	portrait
10	radio	movie

For query word: *program*

Rank	DSIM	DFEAT
1	system	project
2	plan	system
3	project	unit
4	service	status
5	policy	schedule
6	effort*	organization*
7	bill*	activity*
8	company*	plan
9	operation	scheme
10	organization*	policy

fier. Pattern-based features were partially effective when combined with DSIM whereas with DFEAT they were simply redundant.

The impact of this study is that it makes unnecessary to carefully choose similarity measures such as Jaccard's — instead, features can be directly input to supervised learning right after their construction. There are still a great deal of issues to address as the current approach is only in its infancy. For example, the formalization of distributional features requires further investigation. Although we adopted total correlation this time, there can be some other construction methods which show higher performance.

Still, we believe that this is one of the best acquisition performances achieved ever and will be an important step to truly practical lexical knowledge acquisition. Setting our future direction on the completely automatic construction of reliable thesaurus or ontology, the approach proposed here is to be applied to and integrated with various kinds of lexical knowledge acquisition methods in the future.

Acknowledgments

The author would like to thank Assoc. Prof. Katsuhiko Toyama and Assis. Prof. Yasuhiro Ogawa for their kind supervision and advice.

References

Ted Briscoe, John Carroll and Rebecca Watson. 2006. The Second Release of the RASP System. *Proc. of the COLING/ACL 06 Interactive Presentation Sessions*, 77–80.

Collins. 2002. Collins Cobuild Major New Edition CD-ROM. HarperCollins Publishers.

James R. Curran and Marc Moens. 2002. Improvements in automatic thesaurus extraction. In Workshop on Unsupervised Lexical Acquisition. *Proc. of ACL SIGLEX*, 231–238.

Christiane Fellbaum. 1998. *WordNet: an electronic lexical database*, MIT Press.

Maayan Geffet and Ido Dagan. 2004. Feature Vector Quality and Distributional Similarity. *Proc. of COLING 04*, 247–253.

Zellig Harris. 1985. Distributional Structure. Jerrold J. Katz (ed.) *The Philosophy of Linguistics*. Oxford University Press, 26–47.

Marti A. Hearst. 1992. Automatic Acquisition of Hyponyms from Large Text Corpora. *Proc. of COLING 92*, 539–545.

Donald Hindle. 1990. Noun classification from predicate-argument structures. *Proc. of ACL 90*, 268–275.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. *Proc. of COLING/ACL 98*, 786–774.

Shachar Mirkin, Ido Dagan, and Maayan Geffet. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. *Proc. of COLING/ACL 06*, 579–586.

Brian Roark and Eugene Charniak. 1998. Noun phrase cooccurrence statistics for semi-automatic lexicon construction. *Proc. of COLING/ACL 98*, 1110–1116.

Roget. 1995. *Roget's II: The New Thesaurus, 3rd ed.* Houghton Mifflin.

Gerda Ruge. 1997. Automatic detection of thesaurus relations for information retrieval applications. *Foundations of Computer Science: Potential - Theory - Cognition*, LNCS, Volume 1337, 499–506, Springer Verlag.

Rion Snow, Daniel Jurafsly, and Andrew Y. Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems (NIPS) 17*.

Julie Weeds, David Weir and Diana McCarthy. 2004. Characterising Measures of Lexical Distributional Similarity. *Proc. of COLING 04*, 1015–1021.