

Acquiring Hyponymy Relations from Web Documents

Keiji Shinzato Kentaro Torisawa

School of Information Science,

Japan Advanced Institute of Science and Technology (JAIST)

1-1 Asahidai, Tatsunokuchi, Nomi-gun, Ishikawa, 923-1292 JAPAN

{skeiji,torisawa}@jaist.ac.jp

Abstract

This paper describes an automatic method for acquiring hyponymy relations from HTML documents on the WWW. Hyponymy relations can play a crucial role in various natural language processing systems. Most existing acquisition methods for hyponymy relations rely on particular linguistic patterns, such as “*NP such as NP*”. Our method, however, does not use such linguistic patterns, and we expect that our procedure can be applied to a wide range of expressions for which existing methods cannot be used. Our acquisition algorithm uses clues such as itemization or listing in HTML documents and statistical measures such as document frequencies and verb-noun co-occurrences.

1 Introduction

The goal of this work is to become able to automatically acquire hyponymy relations for a wide range of words or phrases from HTML documents on the WWW. We do not use particular lexicosyntactic patterns, as previous attempts have (Hearst, 1992; Caraballo, 1999; Imasumi, 2001; Fleischman et al., 2003; Morin and Jacquemin, 2003; Ando et al., 2003). The frequencies of use for such lexicosyntactic patterns are relatively low, and there can be many words or phrases that do not appear in such patterns even if we look at a large number of texts. The effort of searching for other clues indicating hyponymy relations is thus significant. We try to acquire hyponymy relations by combining three different types of clue obtainable from a wide range of words or phrases. The first type of clue is inclusion in itemizations or lists found in typical HTML documents on the WWW. The second consists of statistical measures such as the document frequency (df) and the inverse document frequency (idf), which are

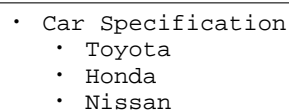
- 
- Car Specification
 - Toyota
 - Honda
 - Nissan

Figure 1: An example of itemization

popular in the IR literature. The third is verb-noun co-occurrence in normal corpora.

In our acquisition, we made the following assumptions.

Assumption A Expressions included in the same itemization or listing in an HTML document are likely to have a common hypernym.

Assumption B Given a set of hyponyms that have a common hypernym, the hypernym appears in many documents that include the hyponyms.

Assumption C Hyponyms and their hypernyms are semantically similar.

Our acquisition process computes a *common hypernym* for expressions in the same itemizations. It proceeds as follows. First, we download a large number of HTML documents from the WWW and extract a set of natural language expressions that are listed in the same itemized region of documents. Consider the itemization in Fig. 1. We extract the set of expressions, {Toyota, Honda, Nissan} from it. From Assumption A, we can treat these expressions as candidates of hyponyms that have a common hypernym such as “company”. We call such expressions in the same itemization *hyponym candidates*. Particularly, a set of the hyponym candidates extracted from a *single* itemization or list is called a *hyponym candidate set (HCS)*. For the example document, we would treat Toyota, Honda, and Nissan as hyponym candidates, and regard them as members of the same HCS.

We then download documents that include at least one hyponym candidate by using an existing search engine, and pick up a noun that appears in the documents and that has the largest *score*. The score was designed so that words appearing in many downloaded documents are highly ranked, according to Assumption B. We call the selected noun a *hypernym candidate* for the given hyponym candidates.

Note that if we download documents including “Toyota” or “Honda”, many will include the word “company”, which is a true hypernym of Toyota. However, words which are not hypernyms, but which are closely associated with Toyota or Honda (e.g., “price”) will also be included in many of the downloaded documents. The next step of our procedure is designed to exclude such non-hypernyms according to Assumption C. We compute the similarity between hypernym candidates and hyponym candidates in an HCS, and eliminate the HCS and its hypernym candidate from the output if they are not semantically similar. For instance, if the previous step of our procedure produces “price” as a hypernym candidate for Toyota and Honda, then the hypernym candidate and the hyponym candidates are removed from the output. We empirically show that this helps to improve overall precision.

Finally, we further elaborate computed hypernym candidates by using additional heuristic rules. Though we admit that these rules are rather ad hoc, they worked well in our experiments.

We have tested the effectiveness of our methods through a series of experiments in which we used HTML documents downloaded from actual web sites. We observed that our method can find a significant number of hypernyms that (at least some of) alternative hypernym acquisition procedures cannot acquire, at least, when only a rather small amount of texts are available.

In this paper, Section 2 describes our acquisition algorithm. Section 3 gives our experimental results which we obtained using Japanese HTML documents, and Section 4 discusses the benefit obtained through our method based on a comparison with alternative methods.

2 Acquisition Algorithm

Our acquisition algorithm consists of four steps, as explained in this section.

Step 1 Extraction of hyponym candidates from itemized expressions in HTML documents.

Step 2 Selection of a hypernym candidate with respect to df and idf.

Step 3 Ranking of hypernym candidates and HCSs based on semantic similarities between hypernym and hyponym candidates.

```
<UL>
  <LI>Car Specification</LI>
  <UL>
    <LI>Toyota</LI>
    <LI>Honda</LI>
    <LI>Nissan</LI>
  </UL>
</UL>
```

Figure 2: An example of HTML documents

Step 4 Application of a few additional heuristics to elaborate computed hypernym candidates and hyponym candidates.

2.1 Step 1: Extraction of hyponym candidates

The objective of Step 1 is to extract an HCS, which is a set of hyponym candidates that may have a common hypernym, from the itemizations or lists in HTML documents. Many methods can be used to do this. Our approach is a simple one. Each expression in an HTML document can be associated with a *path*, which specifies both the HTML tags that enclose the expression and the order of the tags. Consider the HTML document in Figure 2. The expression “Car Specification” is enclosed by the tags ``, `` and ``, ``. If we sort these tags according to their nesting order, we obtain a path (UL, LI) and this path specifies the information regarding the place of the expression. We write $\langle (UL, LI), \text{Car Specification} \rangle$ if (UL, LI) is a path for the expression “Car Specification”. We can then obtain the following paths for the expressions from the document.

```
\langle (UL, LI), Car Specification \rangle,
\langle (UL, UL, LI), Toyota \rangle,
\langle (UL, UL, LI), Honda \rangle,
\langle (UL, UL, LI), Nissan \rangle
```

Basically, our method extracts the set of expressions associated with the same path as an HCS¹. In the above example, we can obtain the HCS {Toyota, Honda, ...}. We extract an itemization only when its size is n and $3 < n < 20$. This is because the processing of large itemizations (particularly the downloading of the related documents) is time-consuming, and small itemizations are often used to obtain a proper layout in HTML documents².

¹We actually need to distinguish different occurrences of the tags in some cases to prevent distinct itemizations from being recognized as a single itemization.

²We found some words that are often inserted into an itemization but do not have common semantic properties with other items in the same itemization, during the experiments using a development set. “リンク (links)” and “ヘルプ (help)” are examples of such words. We prepared a list of such words consisting of 70 items, and removed them from the HCSs obtained in Step 1.

2.2 Step 2: Selection of a hypernym candidate by df and idf

In Step 1, we can obtain a set of hyponym candidates, an HCS, that may have a common hypernym. In Step 2, we select a common hypernym candidate for an HCS. First, we prepare two sets of documents. We randomly select a large number of HTML documents and download them. We call this set of documents a *global document set*. We assume this document set indicates the *general* tendencies of word frequencies. Then we download the documents including each hyponym candidate in a given HCS. This document set is called a *local document set*, and we use it to know the strength of the *association* of nouns with the hyponym candidates.

Let us denote a given HCS as C , a local document set obtained from all the items in C as $LD(C)$, and a global document set as G . We also assume that N is a set of words, which can be candidates of hypernym³. A hypernym candidate, denoted as $h(C)$, for C is obtained through the following formula, where $df(n, D)$ is the number of documents that include a noun n in a document set D .

$$h(C) = \operatorname{argmax}_{n \in N} \{df(n, LD(C)) \cdot idf(n, G)\}$$

$$idf(n, G) = \log \frac{|G|}{df(n, G)}$$

The score has a large value for a noun that appears in a large number of documents in the local document set and is found in a relatively small number of documents in the global document set.

In general, nouns strongly associated with many items in a given HCS tend to be selected through the above formula. Since hyponym candidates tend to share a common semantic property, and their hypernym is one of the words strongly associated with the common property, the hypernym is likely to be picked up through the above formula. Note that a process of *generalization* is performed automatically by treating all the hyponym candidates in an HCS simultaneously. That is, words strongly connected with only one hyponym candidate (for instance, “Lexus” for Toyota) have relatively low score values since we obtain statistical measures from all the local document sets for all the hyponym candidates in an HCS.

Nevertheless, this scoring method is a weak method in one sense. There could be many non-hypernyms that are

³In our experiments, N is a set consisting of 37,639 words, each of which appeared more than 500 times in 33 years of Japanese newspaper articles (Yomiuri newspaper 1987-2001, Mainichi newspaper 1991-1999 and Nikkei newspaper 1983-1990; 3.01 GB in total). We excluded 116 nouns that we observed never be hypernyms from N . An example of such noun is “僕 (I)”. We found them in the experiments using a development set.

strongly associated with many of the hyponym candidates (for instance, “price” for Toyota and Honda). Such non-hypernyms are dealt with in the next step.

An evident alternative to this method is to use $tf(n, LD(C))$, which is the frequency of a noun n in the local document set, instead of $df(n, LD(C))$. We tried using this method in our experiments, but it produced less accurate results, as we show in Section 3.

2.3 Step 3: Ranking of hypernym candidates and HCSs by semantic similarity

Thus, our procedure can produce pairs consisting of a hypernym candidate and an HCS, which are denoted by $\{\langle h(C_1), C_1 \rangle, \langle h(C_2), C_2 \rangle, \dots, \langle h(C_m), C_m \rangle\}$. Here, C_1, \dots, C_m are HCSs, and $h(C_i)$ is a common hypernym candidate for hyponym candidates in an HCS C_i . In Step 3, our procedure ranks these pairs by using the semantic similarity between $h(C_i)$ and the items in C_i . The final output of our procedure is the top k pairs in this ranking after some heuristic rules are applied to it in Step 4. In other words, the procedure discards the remaining $m - k$ pairs in the ranking because they tend to include erroneous hypernyms.

As mentioned, we cannot exclude non-hypernyms that are strongly associated with hyponym candidates from the hypernym candidate obtained by $h(C)$. For example, the value of $h(C)$ may be a non-hypernym “price”, rather than “company”, when $C = \{Toyota, Honda\}$. The objective of Step 3 is to exclude such non-hypernyms from the output of our procedure. We expect such non-hypernyms to have relatively low semantic similarities to the hyponym candidates, while the behavior of *true* hypernyms should be semantically similar to the hyponyms. If we rank the pairs of hypernym candidates and HCSs according to their semantic similarities, the low ranked pairs are likely to have an erroneous hypernym candidate. We can then obtain relatively precise hypernyms by discarding the low ranked pairs.

The similarities are computed through the following steps. First, we parse all the texts in the local document set, and check the argument positions of verbs where hyponym candidates appear. (To parse texts, we use a downgraded version of an existing parser (Kanayama et al., 2000) throughout this work.) Let us denote the frequency of the hyponym candidates in an HCS C occupying an argument position p of a verb v as $f_{hyponym}(C, p, v)$. Assume that all possible argument positions are denoted as $\{p_1, \dots, p_l\}$ and all the verbs as $\{v_1, \dots, v_m\}$. We then define the co-occurrence vector of hyponym candidates as follows.

$$hyponym(C) = \langle f_{hyponym}(C, p_1, v_1), f_{hyponym}(C, p_2, v_1), \dots, f_{hyponym}(C, p_{l-1}, v_m), f_{hyponym}(C, p_l, v_m) \rangle$$

In the same way, we can define the co-occurrence vec-

tor of a hypernym candidate n .

$$\text{hyperv}(n) = \langle f(n, p_1, v_1), \dots, f(n, p_l, v_m) \rangle$$

Here, $f(n, p, v)$ is the frequency of a noun n occupying an argument position p of a verb v obtained from the parsing results of a large number of documents - 33 years of Japanese newspaper articles (Yomiuri newspaper 1987-2001, Mainichi newspaper 1991-1999, and Nikkei newspaper 1990-1998; 3.01 GB in total) - in our experimental setting.

The semantic similarities between hyponym candidates in C and a hypernym candidate n are then computed by a cosine measure between the vectors:

$$\text{sim}(n, C) = \frac{\text{hypov}(C) \cdot \text{hyperv}(n)}{|\text{hypov}(C)| |\text{hyperv}(n)|}$$

Our procedure sorts the hypernym-HCS pairs $\{(h(C_i), C_i)\}_{i=1}^m$ using the value

$$\text{sim}(h(C_i), C_i) \cdot \text{df}(h(C_i), LD(C_i)) \cdot \text{idf}(h(C_i), G)$$

Note that we consider not only the similarity but also the $\text{df} \cdot \text{idf}$ score used in Step 2 in the sorting.

An evident alternative to the above method is the algorithm that re-ranks the top j hypernym candidates obtained by $\text{df} \cdot \text{idf}$ for a given HCS by using the same score. However, we found no significant improvement when this alternative was used in our experiments, as we later explain.

2.4 Step 4: Application of other heuristic rules

The procedure described up to now can produce a hypernym for hyponym candidates with a certain precision. We found, though, that we can improve accuracy by using a few more heuristic rules, which are listed below.

Rule 1 If the number of documents that include a hypernym candidate is less than the sum of the numbers of the documents that include an item in the HCS, then discard both the hypernym candidate and the HCS from the output.

Rule 2 If a hypernym candidate appears as substrings of an item in its HCS and it is not a suffix of the item, then discard both the hypernym candidate and the HCS from the output. If a hypernym candidate is a suffix of its hyponym candidate, then half of the members of an HCS must have the hypernym candidate as their suffixes. Otherwise, discard both the hypernym candidate and its HCS from the output.

Rule 3 If a hypernym candidate is an expression belonging to the category of place names, then replace it by “place name”.

In general, we can expect that a hypernym is used in a wider range of contexts than those of its hyponyms, and that the number of documents including the hypernym candidate should be larger than the number of web documents including hyponym candidates. This justifies Rule 1. We use the hit counts given by an existing search engine as the number of documents including an expression.

As for Rule 2, note that Japanese is a head final language, and a semantic head of a complex noun phrase is the last noun. Consider the following two Japanese complex nouns.

amerika-eiga / nihon-eiga
(American) (movie) / (Japanese) (movie)

Apparently an American movie is a kind of movie as is a Japanese movie. There are many multi-word expressions whose hypernyms are their suffixes, and if some expressions share a common suffix, it is likely to be their hypernym. However, if a hypernym candidate appears in a position other than as a suffix of a hyponym candidate, the hypernym candidate is likely to be an erroneous one. In addition, if a hypernym candidate is a common suffix of only a small portion of an HCS, then the HCS tends not to have semantic uniformity, and such a hypernym candidate should be eliminated from the output. (We empirically determined “one-half” as a threshold in our experiments on the development set.)

As for Rule 3, in our experiments on a development set, we found that our procedure could not provide precise hypernyms for place names such as “Kyoto” and “Tokyo”. In the case of Kyoto and Tokyo, our procedure produced “Japan” as a hypernym candidate. Although “Japan” is consistent with most of our assumptions regarding hypernyms, it is a *holonym* of Kyoto and Tokyo, but their hypernym. In general, when a set of place names is given as an HCS, the procedure tends to produce the name of the region or area that includes all the places designated by the hyponym candidates. We then added the rule to replace such place names by the expression “place name,” which is a *true* hypernym in many of such cases⁴.

Recall that we obtained the ranked pairs of an HCS and its common hypernym in Step 3. By applying the above rules, some pairs are removed from the ranked pairs, or are modified. For some given integer k , the top k pairs of the obtained ranked pairs become the final output of our procedure, as mentioned before.

3 Experimental Results

We downloaded about 8.71×10^5 HTML documents (10.4 GB with HTML tags), and extracted 9.02×10^4 HCSs through the method described in Section 2.1. We

⁴To judge if a hypernym candidate is a place name, we used the output of a morphological analyzer (Matsumoto et al., 1993).

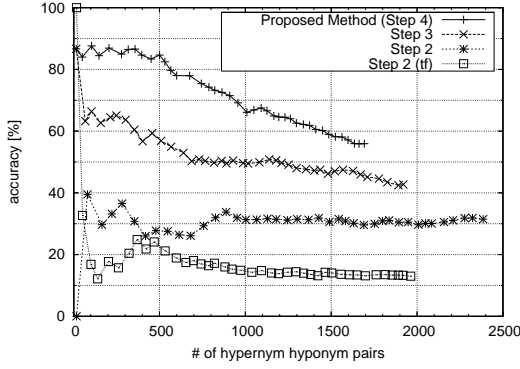


Figure 3: Contribution of each step

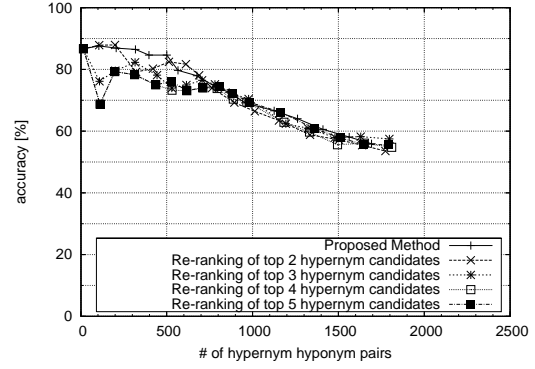


Figure 5: Contribution of re-ranking

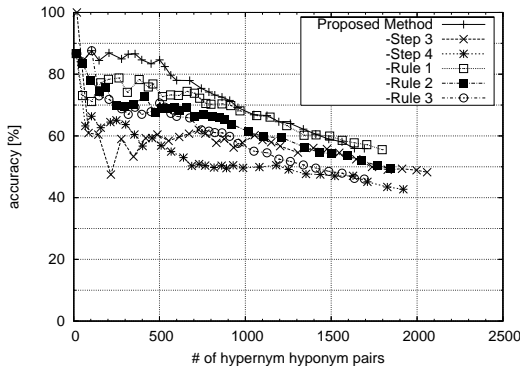


Figure 4: Contribution of each step and rule

randomly picked 2,000 HCSs from among the extracted HCS as our test set. The test set contained 13,790 hyponym candidates. (Besides these HCSs, we used a development set consisting of about 4,000 HCSs to develop our algorithm.) For *each single hyponym candidate*, we downloaded the top 100 documents in the ranking produced by a search engine⁵ as a local document set if the engine found more than 100 documents. Otherwise, all the documents were downloaded. (Note that a local document set for an HCS may contain more than 100 documents.) As a global document set, we used the downloaded 1.00×10^6 HTML documents (1.26 GB without HTML tags).

Fig. 3 shows the accuracy of hypernyms obtained after Steps 2, 3, and 4. We assumed each step produced the sorted pairs of an HCS and a hypernym, which are denoted by $\{\langle h(C_1), C_1 \rangle, \langle h(C_2), C_2 \rangle, \dots, \langle h(C_m), C_m \rangle\}$. The sorting was done by the score $sim(h(C_i), C_i) \cdot df(h(C_i), LD(C_i)) \cdot idf(h(C_i), G)$ after Steps 3 and 4, as described before, while the output of Step 2 was sorted by the $df \cdot idf$ score. In addition, we assumed

⁵The search engine “goo”. (<http://www.goo.ne.jp>)

each step produced only the top 200 pairs in the sorted pairs. (Since the output of Step 4 is the final output, this means that we also assumed that only the top 200 pairs of a hypernym and an HCS would be produced as final output with our procedure. In other words, the remaining 1,800 (=2,000-200) pairs were discarded.)

The resulting hypernyms were checked by the authors according to the definition of the hypernym given in Miller et al., 1990, i.e., we checked if the expression “*a hyponym candidate* is a kind of *a hypernym candidate*.” is acceptable. Then, we computed the precision, which is the ratio of the correct hypernym-hyponym pairs against all the pairs obtained from the top n pairs of an HCS and its hypernym candidate. The x-axis of the graph indicates the number of hypernym-hyponym pairs obtained from the top n pairs of an HCS and its hypernym candidate, while the y-axis indicates the precision.

More precisely, the curve for Step i plots the following points, where $1 \leq j \leq 200$.

$$\left\langle \sum_{k=1}^j |C_k|, \frac{\sum_{k=1}^j correct(C_k, h(C_k))}{\sum_{k=1}^j |C_k|} \right\rangle$$

$correct(C_k, h(C_k))$ indicates the number of hyponym candidates in C_k that are *true* hyponyms of $h(C_k)$. Note that after Step 4, the precision reached about 75% for 701 hyponym candidates, which was slightly more than 5% of all the given hyponym candidates. For 1398 hyponym candidates (about 10% of all the candidates), the precision was about 61%.

Another important point is that “Step 2 (tf)” in the graph refers to an alternative to our Step 2 procedure; i.e., the Step 2 procedure in which $df(h(C), LD(C))$ was replaced by $tf(h(C), LD(C))$. One can see the Step 2 procedure with df works better than that with tf .

Table 1 shows some examples of the acquired HCSs and their common hypernyms. Recall that a common suffix of an HCS is a good candidate to be a hypernym. The examples were taken from cases where a common suffix

hypernym 「hyponym」, hyponym .* 以外の .* hypernym,
 hyponym .* など (、 | の)? hypernym,
 hyponym .* のような .* hypernym,
 hyponym .* に似た .* hypernym,
 hyponym .* と (い | 言) う .* hypernym,
 hyponym .* と呼ばれる .* hypernym,
 hyponym .* (ら | たち) .* hypernym
 The hypernym and hyponym may be bracketed by 「」 or “”.

Figure 6: lexicosyntactic patterns

of an HCS was not produced as a hypernym. This list is actually the output of Step 3, and shows which HCSs and their hypernym candidates were eliminated/modified from the output in Step 4 and which rule was fired to eliminate/modify them.

Next, we eliminated some steps from the whole procedure. Figure 4 shows the accuracy when one of the steps was eliminated from the procedure. “-Step X” or “-Rule X” refers to the accuracies obtained through the procedure from which step X or rule X were eliminated. Note that both graphs indicate that every step and rule contributed to the improvement of the precision.

Figure 5 compares our method and an alternative method, which was the algorithm that re-ranks the top j hypernym candidates for a given HCS by using the score $sim(h, C) \cdot df(h, LD(C)) \cdot idf(h, G)$, where h is a hypernym candidate, in Step 3. (Recall that our algorithm uses the score only for sorting pairs of HCSs and their hypernym. In other words, we do not re-rank the hypernym candidates for a single HCS.) We found no significant improvement when the alternative was used.

4 Comparison with alternative methods

We have shown that our assumptions are effective for acquiring hypernyms. However, there are other alternative methods applicable under our settings. We evaluated the followings methods and compared the results with those of our procedure.

Alternative 1 Compute the non-null suffixes that are shared by the maximum number of hyponym candidates, and regard the longest as a hypernym candidate.

Alternative 2 Extract hypernyms for hyponym candidates by looking at the captions or titles of the itemizations from which hyponym candidates are extracted.

Alternative 3 Extract hypernyms by using lexicosyntactic patterns.

Alternative 4 Combinations of Alternative 1-3.

The evaluation method for Alternative 1 and Alternative 2 is the same as the one for our method. We simply

judged if the produced hypernyms are acceptable or not. But we used different evaluation method for the other alternatives. We checked if the correct hypernyms produced by our method can be found by these alternatives. This is simply for the sake of easiness of the evaluation. Note that we evaluated Alternative 1 and Alternative 2 in the second evaluation scheme when they are combined and are used as a part of Alternative 4.

More detailed explanations on the alternative methods are given below.

Alternative 1 Recall that Japanese is a head final language, and we have explained that common suffixes of hyponym candidates are good candidates to be common hypernyms. Alternative 1 computes a hypernym candidate according to this principle.

Alternative 2 This method uses the captions of the itemizations, which are likely to contain a hypernym of the items in the itemization. We manually found captions or titles that are in the position such that they can explain the content of the itemization, and picked up the caption closest to the itemization and the second closest to it. Then, we checked if the picked-up captions included the proper hypernyms. Note that the precision obtained by this method is just an upper bound of real performance because we do not have a method to *extract* hypernyms from captions at least at the current stage of our research.

Alternative 3 We prepared the lexicosyntactic patterns in Fig. 6, which are similar to the ones used in the previous studies of hypernym acquisition in Japanese (Imasumi, 2001; Ando et al., 2003). One difference from the previous studies was that we used a regular expression instead of a parser. This may have caused some errors, but our patterns were more generous than those used in the previous studies, and did not miss the expressions matched to the patterns from the previous studies. In other words, the accuracy obtained with our patterns was an upper bound on the performance obtained by the previous proposal. Another difference was that the procedure was given *correct* pairs of a hypernym and a hyponym computed beforehand using our proposed method, and it only checked whether given pairs could be found by using the lexicosyntactic patterns from given texts. In other words, this alternative method checked if the lexicosyntactic patterns could find the hypernym-hyponym pairs successfully obtained by our procedure. The texts used were local document sets from which our procedure computed a hypernym candidate. If our procedure has better figures than this method, this means that our procedure can produce hypernyms that cannot be acquired by patterns, at least, from a rather small number of texts (i.e., a maximum of 100 documents per hyponym candidate).

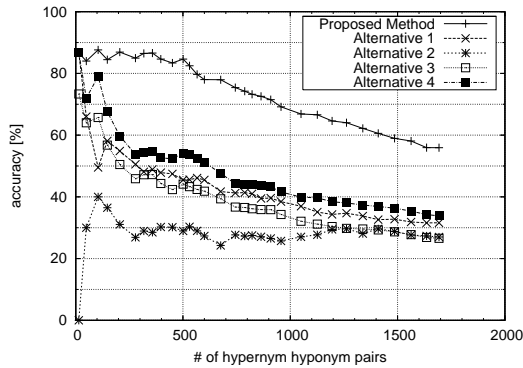


Figure 7: Comparison with alternative methods

Alternative 4 We also compared our procedure with the combination of all the above methods: Alternative 4. Again, we checked whether the combination could find the correct hypernym-hyponym pairs provided by our method. The difference between the precision of our method and that of Alternative 4 reflects the number of hypernym-hyponym pairs that our method could acquire and that Alternative 4 could not. We assumed that for a given HCS a hypernym was successfully acquired if one of the above methods could find the correct hypernym. In other words, the performance of Alternative 4 would be achieved only when there were a technique to combine the output of the above methods in an optimal way.

Figure 7 shows the comparison between our procedure and the alternative methods. We plotted the graph assuming the pairs of hypernym candidates and hyponym candidates were sorted in the same order as the order obtained by our procedure⁶. The results suggest that our method can acquire a significant number of hypernyms that the alternative methods cannot obtain, when we gave rather small amount of texts, a maximum of 100 documents per hyponym candidate, as in our current experimental settings. There is possibility that the difference, particularly the difference from the performance of Alternative 3, becomes smaller when we give more texts to the alternative methods. But the comparison in such settings is actually a difficult task because of the time required for downloading. It is our possible future work.

5 Concluding Remarks and Future Work

We have proposed a method for acquiring hyponymy relations from Web documents, and have shown its effectiveness through experimental results. We also showed

⁶More precisely, we sorted only the hyponym candidates in the order used by our procedure for sorting, and attached the hypernym candidates produced by each alternative to the hyponym candidates.

that our method could find a significant number of hyponymy relations that alternative methods could not, at least when the amount of documents used was rather small.

The first goal of our future work is to further improve the precision of our method. One possible approach will be to combine our methods with alternative techniques, which were actually examined in our experiments. Our second goal is to extend our method so that it can handle multi-word hypernyms. Currently, our method produces just “company” as a hypernym of “Toyota”. If we can obtain a multi-word hypernym such as “automobile manufacturer,” it can provide more useful information to various types of natural language processing systems.

References

- Maya Ando, Satoshi Sekine, and Shun Ishizaki. 2003. Automatic extraction of hyponyms from newspaper using lexicosyntactic patterns. In *IPSJ SIG Technical Report 2003-NL-157*, pages 77–82. in Japanese.
- Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of 37th Annual Meeting of the Association for Computational Linguistics*, pages 120–126.
- Michael Fleischman, Eduard Hovy, and Abdessamad Echihabi. 2003. Offline strategies for online question answering: Answering questions before they are asked. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 1–7.
- Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545.
- Kyosuke Imasumi. 2001. Automatic acquisition of hyponymy relations from coordinated noun phrases and appositions. Master’s thesis, Kyushu Institute of Technology.
- Hiroshi Kanayama, Kentaro Torisawa, Yutaka Mitsuishi, and Jun’ichi Tsujii. 2000. A hybrid Japanese parser with hand-crafted grammar and statistics. In *Proceedings of COLING 2000*, pages 411–417.
- Yuji Matsumoto, Sadao Kurohashi, Takehito Utsuro, Hiroshi Taeki, and Makoto Nagao. 1993. *Japanese Morphological Analyzer JUMAN user’s manual*. in Japanese.
- George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to wordnet: An on-line lexical database. *Journal of Lexicography*, 3(4):235–244.
- Emmanuel Morin and Christian Jacquemin. 2003. Automatic acquisition and expansion of hypernym links. In *Computer and the Humanities 2003*. forthcoming.

Table 1: Examples of the acquired pairs of a hypernym candidate and HCS.

Rank by Step4	Hyponym candidate sets	Hypernyms obtained in Step3	Rank by Step3	Fired Rules			Hypernyms obtained in Step4
				1	2	3	
29	殺人 (murder)*, 放火 (arson)*, 強姦 (rape)*, 侵入盗 (burglary)*, 侵入強盗 (burgle robbery)*, 非侵入盗 (theft without breaking-in)*, 非侵入強盗 (robbery without breaking-in)*	犯罪 (crime)	68	-	-	-	犯罪 (crime)
69	モスクワ (Moscow)*, キエフ (Kiev)*, タシケント (Tashkend)*, ミンスク (Minsk)*, トビリシ (Tbilisi)*, ドウシャンベ (Dushanbe)*, ビシュケク (Bishkek)*, アスタナ (Astana)*, キシニョフ (Kishinev)*, エレバン (Erevan)*, バクー (Baku)*, アシハバード (Ashkhabad)*	ロシア (Russia)	169	-	-	+	地名 (place name)
78	セギノール (Seguignol)*, 藤井康雄 (Yasuo Fujii)*, 五島裕二 (Yuji Goshima)*, 玉木朋孝 (Tomotaka Tamaki)*, 福留宏紀 (Hiroki Fukutome)*, 平野恵一 (Keiichi Hirano)*, シェルドン (Sheldon)*, 塩谷和彦 (Kazuhiko Shiotani)*, (These are baseball players.)	選手 (player)	196	-	-	-	選手 (player)
81	ワイヤレスカード (wireless card), 小電力セキュリティ (security), 市民ラジオ (radio), 特定小電力機器 (a kind of instrument), P H S 陸上移動局 (a kind of department)	無線 (wireless)	200	-	-	-	無線 (wireless)
116	イワウメ (Diapensia lapponica)*, チシマザサ (Sasa kurilensis), キバナシャクナゲ (Rhododendron aureum)*, ミヤマナルコユリ (Polygonatum lasianthum)*	花 (flower)	280	-	-	-	花 (flower)
127	シイタケ (shiitake mushroom)*, サンゴハリタケ (Heridium ramosum)*, サンコタケ (Pseudocolus schellenbergiae)*, シロイボカサタケ (Rhodophyllus murraini)*, シロオニタケ (Amanita virgineoides Bas)*, サンコタケ (Pseudocolus schellenbergiae)*	キノコ (mash-room)	306	-	-	-	キノコ (mash-room)
139	音楽 (music), 映画 (movie), マンガ (cartoon), 出会い (encounter), 芸能人 (artiste)	サイト (web site)	324	-	-	-	サイト (web site)
150	芥川竜之介 (Ryunosuke Akutagawa), 鷹野つぎ (Tsugi Takano), 若山牧水 (Bokusui Wakayama), 梶井基次郎 (Motojiro Kajii), 徳富蘆花 (Roka Tokutomi), 宮本百合子 (Yuriko Miyamoto), 夏目漱石 (Soseki Natume), 田中真太郎 (Kantaro Tanaka), 国木田独步 (Doppo Kunikida), 夢野久作 (Kiyusaku Yumeno), ブレイクウィリアム (William Blake), 菊池寛 (Kan Kikuchi), 夢野久作海若藍平 (parse error) (These are novelists.)	作品 (work)	343	-	-	-	作品 (work)
172	メーデー (May Day), クリスマス (Christmas Day), イースター (Easter), 新年 (the New Year), 万聖節 (All Saints' Day), 主顕節 (Epifania), 解放記念日 (Emancipation Day), 聖母受胎祭 (Immacolata concezione), 聖ステファノの日 (Stefano's Day), 聖母昇天祭 (Ferragosto) (These are national holidays in Italy.)	日本 (Japan)	391	-	-	+	地名 (place name)
184	おかあさん (mother)*, 暖流 (warm current)*, 浮雲 (cloud drift)*, 青空娘 (blue sky girl)*, 美貌に罪あり (beauty has guilt)* (These are Japanese movies.)	映画 (movie)	416	-	-	-	映画 (movie)
-	銀河群 (group of galaxies), 構成メンバー (member), アンドロメダ銀河 (Andromeda Galaxy)*, 銀河系 (The Galaxy)*, 局部銀河群 (local group of galaxies)	銀河 (galaxy)	10	-	+	-	-
-	ブラジル (Brazil), フィリピン (Philippine), 韓国 (Korea), インド (India), アメリカ (U.S.A.), タイ (Thailand), 中国 (China), ペルー (Peru), オーストラリア (Australia), アルゼンチン (Argentina), スペイン (Spain)	日本 (Japan)	80	+	-	+	-

*' indicates a hyponym candidate that is a true hyponym of the provided hypernym candidate.

'+' in the "Fired Rules" column indicates a firing rule, while '-' specifies the rule that doesn't fire.