Extracting Hyponyms of Prespecified Hypernyms from Itemizations and Headings in Web Documents

Keiji Shinzato **Kentaro Torisawa**
Japan Advanced Institute of Science and Technology,
1-1 Asahidai, Tatsunokuchi-machi, Nomi-gun, Ishikawa, 923-1292 JAPAN
{skeiji,torisawa}@jaist.ac.jp

Abstract

This paper describes a method to acquire hyponyms for given hypernyms from HTML documents on the WWW. We assume that a heading (or explanation) of an itemization (or listing) in an HTML document is likely to contain a hypernym of the items in the itemization, and we try to acquire hyponymy relations based on this assumption. Our method is obtained by extending Shinzato's method (Shinzato and Torisawa, 2004) where a common hypernym for expressions in itemizations in HTML documents is obtained by using statistical measures. By using Japanese HTML documents, we empirically show that our proposed method can obtain a significant number of hyponymy relations which would otherwise be missed by alternative methods.

1 Introduction

Hyponymy relations can play a crucial role in various NLP systems, and there have been many attempts to develop automatic methods to acquire hyponymy relations from text corpora (Hearst, 1992; Caraballo, 1999; Imasumi, 2001; Fleischman et al., 2003; Morin and Jacquemin, 2003; Ando et al., 2003). Most of these techniques have relied on particular linguistic patterns, such as "NP such as NP." The frequencies of use for such linguistic patterns are relatively low, though, and there can be many expressions that do not appear in such patterns even if we look at large corpora. The effort of searching for other clues indicating hyponymy relations is thus significant.

Our aim is to extract hyponyms of prespecified hypernyms from the WWW. We use itemizations (or lists) in HTML documents, such as the one in Figure 1(A), and their headings ('Car Company List' in the figure). In a similar attempt, Shinzato and Torisawa proposed an automatic method to obtain a common hypernym of expressions in the same itemizations in HTML documents (Shinzato and Torisawa, 2004) by using statistical measures such as document frequencies and inverse document frequencies. In the following, we call this method the *Algorithm for Hyponymy Relation Acquisition from Itemizations (AHRAI)*. On the other hand, the method we propose in this paper is called *Hyponym Extraction Algorithm from Itemizations and Headings (HEAIH)*.

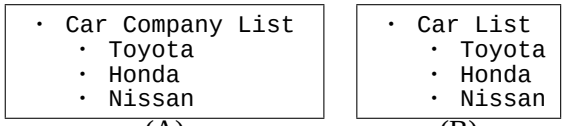

Figure 1: Examples of itemization

The difference between AHRAI and HEAIH is that HEAIH uses the headings attached to itemizations, while AHRAI obtains hypernyms without looking at the headings. This difference has a significant consequence in the acquisition of hyponymy relations. A hyponym tends to have more than one hypernym. For instance, "Toyota" can have at least two hypernyms, "company" and "car." AHRAI may be able to obtain "company," for instance, from the itemizations presented in Figure 1(A), but it cannot simultaneously obtain "car." Consider the itemization in Figure 1(B). Although the heading of the itemization says that the items in the itemizations are cars, AHRAI will provide "company" as a hypernym of the itemizations. This is because AHRAI does not use the headings as clues for finding hypernyms and the itemizations in (A) and (B) are actually identical. Of course, the method could produce the hypernym "car" from different itemizations; this is unlikely, though, because the itemizations *suggesting* that "Toyota" is a "car" are likely to again include the names of other car manufactures such as "Nissan" and "Honda," so the itemization must be more or less similar to the ones in the figure. In such situations, the procedure is likely to consistently produce "company" instead of "car."

On the other hand, HEAIH can simultaneously recognize "Toyota" as a hyponym of the two hypernyms by using the headings. Given a set of hypernyms, for which we'd like to acquire their hyponyms, HEAIH finds the headings (or, more precisely, candidates of headings) that include the given hypernyms, and extracts the itemizations which are located near the headings. The procedure then produces hyponymy relations under the assumption that the expressions in the itemizations are hyponyms of the given hypernym. For example, given "car" and "car company" as hypernyms, the procedure finds

documents including headings such as "Car Company List" and "Car List." If it is lucky enough, it finds documents such as those in Figure 1, and extracts the expressions "Toyota" "Honda," and "Nissan" from the itemizations near the headings. It will then obtain that "Toyota" is a hyponym of "car company" from document (A) in the figure, while it finds that "Toyota" is a hyponym of "car" from (B).

However, the task is not that simple. A problem is that we do not know how to identify the correspondence between itemizations and their headings precisely. One may think that, for instance, she can use the *distance* between an itemization and (candidates of) its heading in the HTML file as a clue for finding the correspondence. However, we empirically show that this is not the case. To solve this problem, we used a modified version of AHRAI. This method can produce a ranked list of hypernym candidates from the itemizations only. We assume that if the top n elements of a ranked list produced by AHRAI include many substrings of a given hypernym, the heading including the hypernym is attached to the itemization.

Note that the original AHRAI produced the top element of the ranked list as a hypernym, while HEAIH recognizes a string as a hypernym if its substrings are included in the top n elements in the ranked list. This helps the HEAIH to acquire hyponymy relations that the AHRAI cannot. Consider the itemizations in Figure 1. AHRAI may produce "company" as the top element of a ranked list for both (A) and (B). But if "car" is in the top n elements in the list as well, HEAIH can recognize "car" as a hypernym for (B).

This paper is organized as follows. Section 2 describes AHRAI. Our proposed method, HEAIH, is presented in Section 3. The experimental results obtained by using Japanese HTML documents are presented in Section 4, where we compared our method and alternative methods.

2 Previous Work: AHRAI

The Algorithm for Hyponymy Relation Acquisition from Itemization (AHRAI) acquires hyponymy relations from HTML documents according to three 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.

We call expressions in an itemization *hyponym candidates*. A set of the hyponym candidates extracted from a *single* itemization or list is called a *hyponym candidate set (HCS)*. For the itemization in Figure 1 (A), we would treat Toyota, Honda, and Nissan as hyponym candidates, and regard them as members of the same HCS.

The procedure consists of the following four steps. Note that Steps 1-3 correspond to Assumptions A-C.

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

Step 2 Selection of a hypernym candidate by using document frequencies and inverse document frequencies.

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

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

Step 1 is performed by using a rather simple algorithm operating on HTML tags. See Shinzato and Torisawa, 2004, for more details. The other steps are described in the following.

2.1 Step 2

In Step 2, the procedure selects a common hypernym candidate for an HCS. First, two sets of documents are prepared. The first set of documents is a large number of HTML documents that are randomly selected and downloaded. This set of documents is called a *global document set*, and is assumed to indicate the *general* tendencies of word frequencies. Then the procedure downloads the documents including each hyponym candidate in a given HCS by using an existing search engine [1]. This document set is called a *local document set*, and is used to determine 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. N is a set of the nouns that can be hypernym candidates[2] A hypernym candidate, denoted as $h(C)$, for C is obtained through the following formula.

$$h(C) = argmax_{n \in N}\{hS(n, C)\}$$

$$hS(n, C) = df(n, LD(C)) \cdot idf(n, G)$$

$df(n, D)$ is a document frequency, which is actually the number of documents including a noun n in a document set D. $idf(n, G)$ is an inverse document frequency, which is defined as $\log(|G|/df(n, G))$.

[1] As in Shinzato and Torisawa, 2004, we used the search engine "goo." (http://www.goo.ne.jp). Note that we enclosed the strings to be searched by "" so that the engine does not split them to words automatically.

[2] We simply used the most frequent nouns observed in a large corpora as N.

The score hS 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. This reflects Assumption B given above.

2.2 Step 3

By Step 2, the procedure can produce pairs of a hypernym candidate and an HCS, which are denoted by $\{\langle h(C_i), C_i\rangle\}_{i=1}^{m}$. Here, C_i is an HCS, and $h(C_i)$ is a common hypernym candidate for hyponym candidates in an HCS C_i.

In Step 3, the similarity between hypernym candidates and hyponym candidates is considered to exclude non-hypernyms that are strongly associated with hyponym candidates from the hypernym candidates obtained by $h(C)$, according to Assumption C. For instance, non-hypernym "price" may be a value of $h(\{Toyota, Honda\})$ because it is strongly associated with the words Toyota and Honda in HTML documents. Such non-hypernyms are excluded based on the assumption that non-hypernyms have relatively low semantic similarities to the hyponym candidates, while the behavior of *true* hypernyms should be semantically similar to the hyponyms. In the "price" example, the similarity between "price" and "Toyota" is relatively low, and we can expect that "price" is excluded from the output.

The semantic similarities between hyponym candidates in an HCS C and a hypernym candidate n are computed using a cosine measure between co-occurrence vectors:

$$sim(n, C) = (ho(C) \cdot hy(n))/(|ho(C)||hy(n)|).$$

Here, $ho(C)$ denotes a co-occurrence vector of hyponym candidates, while $hy(n)$ is the co-occurrence vector of a hypernym candidate n. Assume that all possible argument positions are denoted as $\{p_1, \cdots, p_l\}$ and $\{v_1, \cdots, v_o\}$ denotes a set of verbs. Then, the above vectors are defined as follows.

$$ho(C) = \langle f_h(C, p_1, v_1), \cdots, f_h(C, p_l, v_o)\rangle$$

$$hy(n) = \langle f(n, p_1, v_1), \cdots, f(n, p_l, v_o)\rangle$$

Here, $f_h(C, p, v)$ denotes the frequency of the hyponym candidates in an HCS C occupying an argument position p of a verb v in a local document set and $f(n, p, v)$ is the frequency of a noun n occupying an argument position p of a verb v in a large document set.

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

$$sim(h(C_i), C_i) \cdot hS(h(C_i), C_i).$$

Then, the top elements of the sorted pairs are likely to contain a hypernym candidate and an HCS that are semantically similar to each other. The final output of AHRAI is the top k pairs in this ranking after some heuristic rules are applied to it in Step 4.

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." Recognition of place names was done by an existing morphological analyzer.

Figure 2: Heuristic rules of AHRAI

In other words, the procedure discards the remaining $m - k$ pairs in the ranking because they tend to include erroneous hypernyms.

2.3 Step 4

The steps described up to now can produce a hypernym for hyponym candidates with a certain precision. However, Shinzato et al. reported that the rules shown in Figure 2 can contribute to higher accuracy. 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. Rule 2 is effective since Japanese is a head final language, and semantic head of a complex noun phrase is the last noun. Rule 3 was justified by the observation that when a set of place names is given as an HCS, the procedure tends to produce the name of the region that includes all the places designated by the hyponym candidates. (See Shinzato and Torisawa, 2004 for more details.)

Recall that in Step 3, the ranked pairs of an HCS and its common hypernym are obtained. By applying the above rules to these, 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 Proposed Method: HEAIH

Our proposed method, Hyponym Extraction Algorithm from Itemizations and Headings (HEAIH), is obtained by using some steps of AHRAI. The HEAIH procedure is given a set of l hypernyms, denoted by $X = \{x_i\}_{i=1}^{l}$, where x_i is a hypernym, and finds hyponyms for the hypernyms. The basic behavior of the HEAIH is summarized as follows. First, it downloads the documents which are likely to contain itemizations consisting of hyponyms of the given hypernyms. This is done by generating possible headings or explanations of the itemizations by using prespecified linguistic patterns and by search-

"X(の) 一覧" (table of X) "X(の) ガイド" (guide to X)
"X(の) カテゴリ" (category of X) "X(の) リスト" (list of X)
"X(の) 投票" (vote to X) "X(の) メニュー" (menu of X)
"X(の) ランキング" (ranking of X)
X is a place holder that a given hypernym fills in.

Figure 3: Patterns for generating headings

ing the documents including the expressions with an existing search engine. Second, the procedure applies Steps 1 and 2 of AHRAI and computes a ranked list of hypernym candidates for each HCS extracted from the itemizations in the downloaded documents. The list is ranked in descending order of the hS score values. Note that the ranked list is generated independently from a given hypernym.

We assume that a given hypernym is likely to be a *true* hypernym if the top elements of the ranked list of hypernym candidates contain many substrings of the hypernym. The procedure computes a score value, which is designed so that it has a large value when many substrings of the given hypernym are included in the list. Then, the pairs of a given hypernym and a corresponding HCS are sorted by the score value, and only the top k pairs are provided as the output of the whole procedure.

More precisely, HEAIH consists of Steps A-E, each of which are described below.

Step A For each of the given hypernyms, denoted by x_i, generate a set of strings which are typically used in headings, such as "List of x_i," by using the prespecified patterns listed in Figure 3. The set of generated strings for a hypernym x_i is denoted by $Hd(x_i)$. Give each string in $Hd(x_i)$ to an existing search engine and pick up a string that has the maximum hit count in $Hd(x_i)$. Then, download the documents in the ranking produced by the engine for the picked up string. In our experiments, we downloaded the top 25 documents for each hypernym if the ranking contained more than 25 documents. Otherwise, all the documents were downloaded.

Step B Identify the itemizations in the downloaded documents and extract the expressions in them by using Step 1 of AHRAI. The results obtained in this step are denoted by $B(X) = \{\langle x'_h, C_h\rangle\}_{h=1}^{m}$, where x'_h is one of the given hypernyms and C_h is an HCS extracted from a document downloaded for x'_h.

Step C Apply Step 2 of AHRAI to each HCS C_h such that $\langle x'_h, C_h\rangle \in B(X)$, and then obtain a ranked list that contains the top p words according to the hS values and is ranked by the values. We denote the list as $HCList(C_h)$.

Step D Sort the set $B(X) = \{\langle x'_h, C_h\rangle\}_{h=1}^{m}$ in descending order of the hSC value, which is given below.

$$hSC(x'_h, C_h) = sim(x'_h, C_h) \cdot \sum_{j=1}^{p} \{sub(x'_h, jth(HCList(C_h), j)) \cdot hS(jth(HCList(C_h), j), C_h)\}$$

$jth(list, j)$ denotes the j-th element in $list$ and

$$sub(x, y) = \begin{cases} 1 & \text{if } y \text{ is a substring of } x \\ 0 & \text{otherwise.} \end{cases}$$

In short, the score hSC is the sum of the score values hS for the substrings of a given hypernym that was contained in the top p elements of the ranked list produced by Step 2 of AHRAI. In our experiments, we assumed $p = 10$. In addition, the score is weighted by the similarity measure $sim(x, C)$[3].

Step E Apply Rules 1 and 2 of AHRAI to each element of the sorted list obtained in Step D, and produce the top k pairs that survived the check by the rules as the final output. In our experiments, we assumed $k = 200$, while we obtained $B(X)$ consisting of 2,034 pairs.

Note that the weighting factor $sim(x, C)$ in hSC contributed to high accuracy in our experiments using a development set.

4 Experimental Results

To evaluate our procedure, we had to provide a set of proper hypernyms for which HEAIH would find hyponyms. This was a rather difficult task. There are many nouns that cannot be hypernyms. We assumed that the Japanese noun sequences or nouns that occupied the position of X in the patterns "X 一覧" (table of X) "X の紹介" (guide to X) "歴代の X" (successive (or chronological list of) X) and "有名 X" (well-known X) in corpora were appropriate as hypernyms. (Despite this filtering, there were some inappropriate hypernyms in the set of hypernyms subjected to the procedures in our experiments. These inappropriate hypernyms included expressions whose hyponyms change drastically according to the situation in which the expressions are used. Examples are "recommended products." One cannot determine the possible hyponyms without knowing *who* is recommending. We judged any hyponymy relations including such hypernyms as being unacceptable.)

We downloaded 1.00×10^6 Japanese HTML documents (1.26 GB without tags), applied the above patterns and found 8,752 expressions. Then, we randomly picked out 100 hypernym candidates from 869 expressions that occurred with the above patterns more than three times, and 100 hypernym candidates from the remaining 7,883 expressions. These 200 hypernym candidates became the input for our procedure. As mentioned, we downloaded a maximum of 25 pages for each hypernym, and extracted

[3] In HEAIH, the hypernym x may not be included in the set of nouns for which we obtained a co-occurrence vector since x is simply given to the procedure from outside, and the procedure may not be able to compute the sim values. In that case, we replace x with the longest suffix of x that is contained in the set of nouns for which co-occurrence vectors were obtained. The head final characteristic of the Japanese language justifies this replacement.

研究室 (laboratories, 34)*, 健康食品 (health food/beverage, 18)*, 福祉施設 (welfare facilities, 13)*, 機能 (functionalities, 12), 都市公園 (parks in cities, 10)*, 店 (stores/shops, 10)*, 皇帝 (emperors, 7)*, 地区 (districts, 6)*, 事業 (businesses, 6), 遺産 (legacies, 6)*, 取り扱い商品 (offered products, 5), 参加企業 (participant companies, 5), 作品 (works of art, 5)*, パーツ (parts of machines, 5), 日本三大○○ (Japan's top three something, 4), 小説 (novels, 4)*, 部活動 (club activities, 3)*, 占いサイト (fortune telling websites, 3)*, 事業制度 (rules of business, 3), タイムアタック (time attack, 3), コマンド (commands, 3), 注目商品 (recommended products, 2), 生産者 (producers,2), 詩 (poems, 2)*, 市 (cities or markets, 2)*, 高山植物 (alpine plants, 2)*, チーム名 (names of teams, 2)*, サイドビジネス (side businesses, 2), お仕事 (jobs, 2), 物件 (things, 1), 日本語版 (Japanese versions, 1), 動物 (animals, 1)*, 専門 (specialties, 1), 紹介 (introductions, 1), 小説家 (novelists, 1)*, 質問 (questions, 1), 資料 (data, 1), 在宅ビジネス (working at home, 1), 学童クラブ (students' clubs, 1)*, 会場 (venues, 1) 駅名 (names of railway stations, 1)*, マルチメディア科 (dept. of multimedia), パワーストーン ("Power Stone" amulet, 1)*, バンド (bands/groups of musicians, 1)*, シェフ (chefs, 1) *, ゲームソフト (game programs, 1)*, キャラ (characters in games/movies/stories, 1)*, アイドル (idols, 1)*,

Figure 4: List of hypernyms in the HEAIH output

3,211 itemizations from them. (We restricted the itemizations to the ones containing less than or equal to 30 items.) Then, we picked out 2,034 itemizations and used them in our evaluation. The choice was made in the following manner. First, for each hypernym candidate, the itemizations were sorted in ascending order of the distance between the occurrence of the hypernym candidate and the itemization in the downloaded page. Then, the itemizations in the top 65% were chosen for each hypernym.[4]. This selection was made to eliminate the itemizations located extremely far from the given hypernyms and to keep the number of itemizations close to 2,000, which was the number of itemizations used in Shinzato and Torisawa, 2004.

Recall that HEAIH (and AHRAI) require two different types of document sets: global document sets and local document sets. As a global document set, we used the downloaded 1.00×10^6 HTML documents used to obtain hypernyms given to the HEAIH. As a local document set for each hyponym candidate, we downloaded the top 100 documents in the ranking produced by a search engine. In addition, we used 5.72×10^6 Japanese HTML documents (6.27 GB without tags) to obtain co-occurrence vectors to calculate the semantic similarities between expressions. To derive co-occurrence vectors, we parsed the documents by using a downgraded version of an existing parser (Kanayama et al., 2000) and collected co-occurrences from the parsing results.

As mentioned, we obtained 200 pairs of a hypernym and an HCS as the final HEAIH output. All the hypernyms appearing in the output are listed in Figure 4 along with their English translations and

[4]Particularly, when only one itemization was obtained for a hypernym, it was selected.

hypernym	HCS
皇帝 (emperor)	*世宗, *始祖, *敬宗, *統宗, *高祖, *恭宗 (These are Chinese Emperors.)
福祉施設 (welfare facilities)	*身体障害者授産施設, *身体障害者療護施設, *重度身体障害者更生施設 (These are welfare facilities)
健康食品 (health food/ beverage)	*ルイボスティー, *プーアル茶, *シモン茶, *グルコケア, *紫イペー (These are teas which are good for health.)
占いサイト (fortune telling websites)	*占いカフェ, *占い比較市場 *矢萩予言研究所, *うらないサーチ (These are fortune telling websites.)
小説家 (novelist)	武揚伝, 田端文士村, 由布院心中事件 (These are novels.)

Figure 5: Examples of the acquired hyponymy relations

the number of HCSs that the procedure produced with the hypernym. In the 200 pairs, 48 hypernyms appeared. The HCSs were taken from 119 distinct websites, and the maximum number of the HCSs taken from a single site was 7. The resulting pairs of hypernym candidates and hyponym candidates 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. Figure 5 shows some examples of the hypernym-HCS pairs that were obtained by HEAIH. A hyponym candidates in the HCSs is marked by "*" if it is a proper hyponym of the hypernym in the pair. We then computed the precision, which was the ratio of correct hypernym-hyponym pairs against all the pairs obtained from the top 200 pairs of an HCS and its hypernym candidate. The graph in Figure 6 plots the precision obtained by HEAIH, along with the precisions of the alternative methods as we explain later. The x-axis of the graph indicates the number of hypernym-hyponym pairs obtained from the top j pairs of an HCS and its hypernym candidate, while the y-axis indicates the precision. More precisely, the curve plots the points denoted by $\langle \sum_{h=1}^{j} |C_i|, (\sum_{h=1}^{j} correct(C_h, x'_h))/(\sum_{h=1}^{j} |C_h|) \rangle$, where the output of the HEAIH is denoted by $\{\langle x'_h, C_h \rangle\}_{h=1}^{200}$ and $1 \leq j \leq 200$. $correct(C_h, x'_h)$ indicates the number of hyponym candidates in C_h that are *true* hyponyms of the hypernym x'_h.

We compared the performances of the following five alternative methods with that of HEAIH.

Alternative 1 Produce pairs consisting of a given hypernym and a hyponym candidate in an HCS if the given hypernym is a suffix of the hyponym candidate. Note that Japanese is a head final language and that suffixes of hyponym candidates are good candidates to be hypernyms.

Alternative 2 Extract hyponymy relations by applying lexicosyntactic patterns to the documents in the local document sets for our method. We used

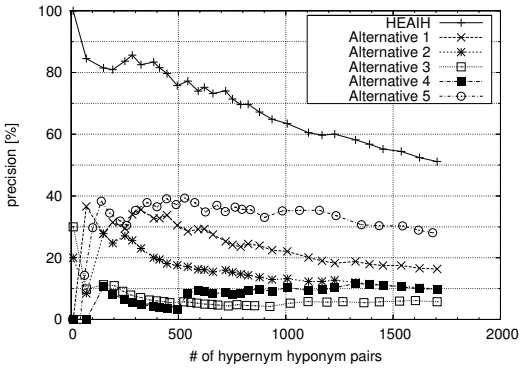

Figure 6: Precision of hyponymy relations

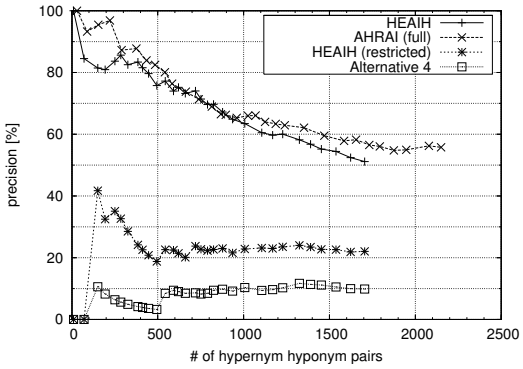

Figure 8: Comparison between HEAIH and AHRAI

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

Figure 7: Lexicosyntactic patterns

patterns proposed in previous work (Imasumi, 2001; Ando et al., 2003) (Figure 7). Note that these are regular expressions and may *overgenerate* hyponymy relations; however, they do not miss the relations acquired through more sophisticated methods such as those with parsers.

Alternative 3 Extract hyponymy relations by looking for lexicosyntactic patterns with an existing search engine. The patterns used were basically the same as those used in Alternative 2. However, the expression ".*" was eliminated from the patterns and the disjunctions "|" were expanded to simple strings since the engine would not accept regular expressions. In addition, the pattern "hypernym「hyponym」" was not used because the brackets "「」" were not treated properly by the engine.

Alternative 4 Original AHRAI.

Alternative 5 Produce hypernym-hyponym pairs according to only the *distance* between the headings including the hypernym and the itemizations including HCSs. Recall that $Hd(x)$ is the set of strings likely to be headings of itemizations for a given hypernym x. This alternative method computes the distance in bytes between the position of a member of $Hd(x)$ in a downloaded document and the position of the itemization including an HCS. The pairs of an itemization and a given hypernym are then sorted according to this distance to produce the 200 pairs with the smallest distance as pairs of hypernyms and the corresponding HCSs. Note that we assumed a heading must appear *before* an HCS.

We checked if the above alternatives can acquire the *correct* pairs of a hypernym and a hyponym obtained by HEAIH. In other words, we counted how many correct pairs produced by HEAIH were also acquired by the alternatives when using the same document set. Note that all the alternative methods except for Alternative 5 were applied only to the 200 pairs of a hypernym and an HCS that were the final HEAIH output. The results are presented in Figure 6. The curves indicate the ratios of correct hyponymy relations that are acquired by an alternative against all the relations produced by HEAIH. As for Alternatives 1-4, 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 procedure. In the case of Alternative 5, the 2,034 pairs of a hypernym candidates and an HCS, which were the results of Step B in HEAIH, were sorted according to the distance between headings and itemizations, and only the top 200 pairs were produced as the final output. The results suggest that our method can acquire a significant number of hyponymy relations that the alternatives miss.

We then conducted a *fairer* comparison between HEAIH and Alternative 4 (or AHRAI). There are some hypernyms that can never be produced by AHRAI since these hypernyms are not considered in AHRAI. Recall that we computed the score hS for the nouns in a set N, which contained the 155,345 nouns most frequently observed in the downloaded 5.72×10^6 documents in our experiments. If a given hypernym was not included in N, AHRAI could not produce that hypernym. In addition, some of the given hypernyms are actually noun sequences (or complex nouns) and cannot be members of N. On the other hand, HEAIH can acquire a hypernym not included in N if the hypernym contains substrings included in N. Thus, we also compared the performance under the assumption that only the hypernyms included in N could be *true* hypernyms. The results are presented in Figure 8. "Alternative 4" refers to the performance of AHRAI, while "HEAIH (restricted)" indicates the performance of HEAIH

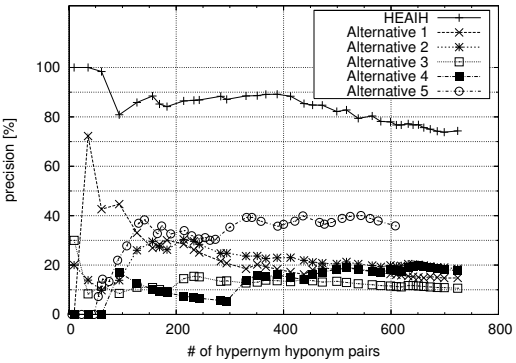

Figure 9: Comparison with balanced data

when the produced hypernyms were restricted to the members of N. They show that HEAIH still outperformed AHRAI. In addition, the curve "AHRAI (full)" shows the performance of AHRAI when we accept the hypernyms that were not given to the HEAIH and all the 2,034 pairs of a hypernym candidate and an HCS were sorted according to the original score for AHRAI to produce the top 200 pairs. In this case, AHRAI outperformed HEAIH, though the difference is small.

In the next set of experiments, we compared HEAIH and Alternatives 1-5 in a slightly different setting. Recall that Figure 4 gave the list of hypernyms in the HEAIH output and the number of HCSs that the procedure produced with each hypernym. The data was not balanced very evenly. While the procedure found 34 HCSs for laboratories, it provided only one HCS for animals. We tried to reevaluate these methods by using more balanced data. From the data, we eliminated the pairs of a hypernym and an HCS that were not included in the top five for each hypernym in the ranking of the HEAIH output. In other words, each hypernym could have a maximum of only five HCSs in the evaluation data. This reduced the influence by *dominant hypernyms*.

In addition, we removed problematic hypernyms from the evaluation data. The preserved hypernyms are marked by '*' in Figure 4. We preserved only the hypernyms that could have proper nouns, names of species, or trade names as their hyponyms.[5] In addition, there are inappropriate hypernyms such as those for which we could not determine their hyponyms without knowing the situation in which the hypernyms are used, as mentioned before. We eliminated such hypernyms too. We also removed "things" because it was too general. As a result of these changes, the evaluation data contained 73 pairs of a hypernym and an HCS. The comparison using this data is shown in Figure 9. HEAIH still acquired a large number of correct hyponymy relations that the alternative methods miss.

[5]Evidently, this condition was more restrictive than we expected with regard to hypernyms, and some intuitively acceptable hypernyms were not preserved. Examples are "jobs" and "business" (For their Japanese translation, we could not find hyponyms which were either proper nouns, names of species, or trade names). We made this restriction simply to keep the condition simple and to reduce *borderline cases* of proper hypernyms. Note that some of the eliminated hypernyms, such as "jobs" and "business", were treated as proper hypernyms in the first comparison in Figure 6.

5 Conclusions

We have presented a new method for acquiring hyponyms for prespecified hypernyms by using itemizations and their headings (or explanations.) This method was developed by modifying Shinzato's algorithm to find hypernyms from itemizations in HTML documents. The method could find a large number of hyponymy relations that alternative methods, including the original Shinzato algorithm, could not.

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 acquistition 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.

Emmanuel Morin and Christian Jacquemin. 2003. Automatic acquisition and expansion of hypernym links. In *Computer and the Humanities 2003*. forthcoming.

Keiji Shinzato and Kentaro Torisawa. 2004. Acquiring hyponymy relations from web documents. In *Proceedings of HLT-NAACL 2004*, pages 73–80.