Co-STAR: A Co-training Style Algorithm for Hyponymy Relation Acquisition from Structured and Unstructured Text

Jong-Hoon Oh, Ichiro Yamada, Kentaro Torisawa, and Stijn De Saeger
Language Infrastructure Group, MASTAR Project,
National Institute of Information and Communications Technology (NICT)
{rovellia,iyamada,torisawa,stijn}@nict.go.jp

Abstract

This paper proposes a co-training style algorithm called Co-STAR that acquires hyponymy relations simultaneously from structured and unstructured text. In Co-STAR, two independent processes for hyponymy relation acquisition – one handling structured text and the other handling unstructured text – collaborate by repeatedly exchanging the knowledge they acquired about hyponymy relations. Unlike conventional co-training, the two processes in Co-STAR are applied to different source texts and training data. We show the effectiveness of this algorithm through experiments on large-scale hyponymy-relation acquisition from Japanese Wikipedia and Web texts. We also show that Co-STAR is robust against noisy training data.

1 Introduction

Acquiring semantic knowledge, especially semantic relations between lexical terms, is regarded as a crucial step in developing high-level natural language applications. This paper proposes Co-STAR (a **Co**-training **ST**yle **A**lgorithm for hyponymy **R**elation acquisition from structured and unstructured text). Similar to co-training (Blum and Mitchell, 1998), two hyponymy relation extractors in Co-STAR, one for structured and the other for unstructured text, iteratively collaborate to boost each other's performance.

Many algorithms have been developed to automatically acquire semantic relations from structured and unstructured text. Because term pairs are encoded in structured and unstructured text in different styles, different kinds of evidence have been used for semantic relation acquisition:

Evidence from unstructured text: lexico-syntactic patterns and distributional similarity (Ando et al., 2004; Hearst, 1992; Pantel et al., 2009; Snow et al., 2006; De Saeger et al., 2009; Van Durme and Pasca, 2008);

Evidence from structured text: topic hierarchy, layout structure of documents, and HTML tags (Oh et al., 2009; Ravi and Pasca, 2008; Sumida and Torisawa, 2008; Shinzato and Torisawa, 2004).

Recently, researchers have used both structured and unstructured text for semantic-relation acquisition, with the aim of exploiting such different kinds of evidence at the same time. They either tried to improve semantic relation acquisition by putting the different evidence together into a single classifier (Pennacchiotti and Pantel, 2009) or to improve the coverage of semantic relations by combining and ranking the semantic relations obtained from two source texts (Talukdar et al., 2008).

In this paper we propose an algorithm called Co-STAR. The main contributions of this work can be summarized as follows.

- Co-STAR is a semi-supervised learning method composed of two parallel and iterative processes over structured and unstructured text. It was inspired by bilingual co-training, which is a framework for hyponymy relation acquisition from source texts in two languages (Oh et al., 2009). Like bilingual co-training, two processes in Co-STAR operate independently on structured text and unstructured text. These two processes are trained in a supervised manner with their initial training data and then each of them tries to enlarge the existing training data of the other by iteratively exchanging what they

have learned (more precisely, by transferring reliable classification results on common instances to one another) (see Section 4 for comparison Co-STAR and bilingual co-training). Unlike the ensemble semantic framework (Pennacchiotti and Pantel, 2009), Co-STAR does not have a single "*master*" classifier or ranker to integrate the different evidence found in structured and unstructured text. We experimentally show that, at least in our setting, Co-STAR works better than a single "*master*" classifier.

- Common relation instances found in both structured and unstructured text act as a *communication channel* between the two acquisition processes. Each process in Co-STAR classifies common relation instances and then transfers its high-confidence classification results to training data of the other process (as shown in Fig. 1), in order to improve classification results of the other process. Moreover, the efficiency of this exchange can be boosted by increasing the "bandwidth" of this channel. For this purpose each separate acquisition process automatically generates a set of relation instances that are likely to be *negative*. In our experiments, we show that the above idea proved highly effective.

- Finally, the acquisition algorithm we propose is robust against noisy training data. We show this by training one classifier in Co-STAR with manually labeled data and training the other with automatically generated but noisy training data. We found that Co-STAR performs well in this setting. This issue is discussed in Section 6.

This paper is organized as follows. Sections 2 and 3 precisely describe our algorithm. Section 4 describes related work. Sections 5 and 6 describe our experiments and present their results. Conclusions are drawn in Section 7.

2 Co-STAR

Co-STAR consists of two processes that simultaneously but independently extract and classify

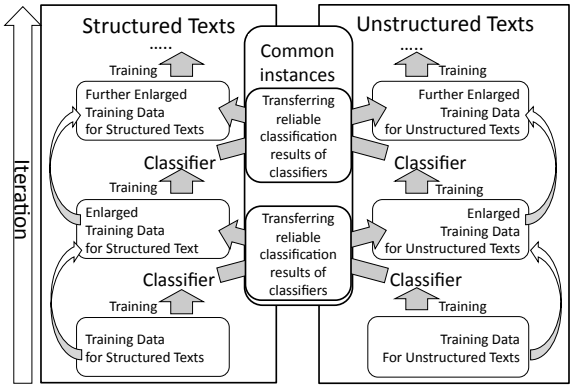


Figure 1: Concept of *Co-STAR*.

hyponymy relation instances from structured and unstructured text. The core of Co-STAR is the collaboration between the two processes, which continually exchange and compare their acquired knowledge on hyponymy relations. This collaboration is made possible through common instances shared by both processes. These common instances are classified separately by each process, but high-confidence classification results by one process can be transferred as new training data to the other.

2.1 Common Instances

Let S and U represent a source (i.e. corpus) of structured and unstructured text, respectively. In this paper, we use the hierarchical layout of Wikipedia articles and the Wikipedia category system as structured text S (see Section 3.1), and a corpus of ordinary Web pages as unstructured text U. Let X_S and X_U denote a set of hyponymy relation candidates extracted from S and U, respectively. X_S is extracted from the hierarchical layout of Wikipedia articles (Oh et al., 2009) and X_U is extracted by lexico-syntactic patterns for hyponymy relations (i.e., *hyponym* such as *hyponymy*) (Ando et al., 2004) (see Section 3 for a detailed explanation)

We define two types of common instances, called "*genuine*" common instances (G) and "*virtual*" common instances (V). The set of common instances is denoted by $Y = G \cup V$. Genuine common instances are hyponymy relation candidates found in both S and U ($G = X_S \cap X_U$). On

the other hand, term pairs are obtained as virtual common instances when:

- 1) they are extracted as hyponymy relation candidates in either S or U and;
- 2) they do not seem to be a hyponymy relation in the other text

The first condition corresponds to $X_S \oplus X_U$. Term pairs satisfying the second condition are defined as R_S and R_U, where $R_S \cap X_S = \phi$ and $R_U \cap X_U = \phi$.

R_S contains term pairs that are found in the Wikipedia category system but neither term appears as ancestor of the other[1]. For example, (*nutrition,protein*) and (*viruses,viral disease*), respectively, hold a category-article relation, where *nutrition* is not ancestor of *viruses* and vice versa in the Wikipedia category system. Here, term pairs, such as (*nutrition, viruses*) and (*viral disease, nutrition*), can be ones in R_S.

R_U is a set of term pairs extracted from U when:

- they are not hyponymy relation candidates in X_U and;
- they regularly co-occur in the same sentence as arguments of the same verb (e.g., A *cause* B or A *is made by* B);

As a result, term pairs in R_U are thought as holding some other semantic relations (e.g., A and B in "A *cause* B" may hold a cause/effect relation) than hyponymy relation. Finally, virtual common instances are defined as:

- $V = (X_S \oplus X_U) \cap (R_S \cup R_U)$

The virtual common instances, from the viewpoint of either S or U, are unlikely to hold a hyponymy relation even if they are extracted as hyponymy relation candidates in the other text. Thus many virtual common instances would be a negative example for hyponymy relation acquisition. On the other hand, genuine common instances (hyponymy relation candidates found in both S and U) are more likely to hold a hyponymy relation than virtual common instances.

In summary, genuine and virtual common instances can be used as different ground for collaboration as well as broader collaboration channel between the two processes than genuine common instances used alone.

[1] A term pair often holds a hyponymy relation if one term in the term pair is a parent of the other in the Wikipedia category system (Suchanek et al., 2007).

2.2 Algorithm

We assume that classifier c assigns class label $cl \in \{yes, no\}$ ("yes" (hyponymy relation) or "no" (not a hyponymy relation)) to instances in $x \in X$ with confidence value $r \in \mathbb{R}^+$, a non-negative real number. We denote the classification result by classifier c as $c(x) = (x, cl, r)$. We used support vector machines (SVMs) in our experiments and the absolute value of the distance between a sample and the hyperplane determined by the SVMs as confidence value r.

The Co-STAR algorithm is given in Fig. 2. The algorithm is interpreted as an iterative procedure 1) to train classifiers (c_U^i, c_S^i) with the existing training data (L_S^i and L_U^i) and 2) to select new training instances from the common instances to be added to existing training data. These are repeated until stop condition is met.

In the initial stage, two classifiers c_S^0 and c_U^0 are trained with manually prepared labeled instances (or training data) L_S^0 and L_U^0, respectively. The learning procedure is denoted by $c = LEARN(L)$ in lines 5–6, where c is a resulting classifier. Then c_S^i and c_U^i are applied to classify common instances in Y (lines 7–8). We denote CR_S^i as a set of the classification results of c_S^i for common instances, which are not included in the current training data $L_S^i \cup L_U^i$. Lines 9–14 describe a way of selecting instances in CR_S^i to be added to the existing training data in U. During the selection, c_S^i acts as a teacher and c_U^i as a student. $TopN(CR_S^i)$ is a set of $c_S^i(y) = (y, cl_S, r_S)$, whose r_S is the top-N highest in CR_S^i. (In our experiments, $N = 900$.) The teacher instructs his student the class label of y if the teacher can decide the class label of y with a certain level of confidence ($r_S > \alpha$) and the student satisfies one of the following two conditions:

- the student agrees with the teacher on class label of y ($cl_S = cl_U$) or

1: **Input**: Common instances ($Y = G \cup V$) and the initial training data (L_S^0 and L_U^0)
2: **Output**: Two classifiers (c_S^n and c_U^n)
3: $i = 0$
4: **repeat**
5: $c_S^i := LEARN(L_S^i)$
6: $c_U^i := LEARN(L_U^i)$
7: $CR_S^i := \{c_S^i(y) | y \in Y, y \notin L_S^i \cup L_U^i\}$
8: $CR_U^i := \{c_U^i(y) | y \in Y, y \notin L_S^i \cup L_U^i\}$
9: $L_U^{(i+1)} := L_U^i$
10: **for** each $(y, cl_S, r_S) \in TopN(CR_S^i)$ and $(y, cl_U, r_U) \in CR_U^i$ **do**
11: **if** ($r_S > \alpha$ and $r_U < \beta$) or ($r_S > \alpha$ and $cl_S = cl_U$) **then**
12: $L_U^{(i+1)} := L_U^{(i+1)} \cup \{(y, cl_S)\}$
13: **end if**
14: **end for**
15: $L_S^{(i+1)} := L_S^i$
16: **for** each $(y, cl_U, r_U) \in TopN(CR_U^i)$ and $(y, cl_S, r_S) \in CR_S^i$ **do**
17: **if** ($r_U > \alpha$ and $r_S < \beta$) or ($r_U > \alpha$ and $cl_S = cl_U$) **then**
18: $L_S^{(i+1)} := L_S^{(i+1)} \cup \{(y, cl_U)\}$
19: **end if**
20: **end for**
21: $i = i + 1$
22: **until** stop condition is met

Figure 2: Co-STAR algorithm

- the student's confidence in classifying y is low ($r_U < \beta$)

$r_U < \beta$ enables the teacher to instruct his student in spite of their disagreement over a class label. If one of the two conditions is satisfied, (y, cl_S) is added to existing labeled instances $L_U^{(i+1)}$. The roles are reversed in lines 15–20, so that c_U^i becomes the teacher and c_S^i the student.

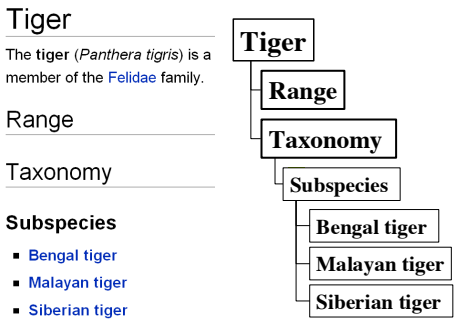
The iteration stops if the change in the difference between the two classifiers is stable enough. The stability is estimated by $d(c_S^i, c_U^i)$ in Eq. (1), where σ^i represents the change in the average difference between the confidence values of the two classifiers in classifying common instances. We terminate the iteration if $d(c_S^i, c_U^i)$ is smaller than 0.001 in three consecutive rounds (Wang and Zhou, 2007).

$$d(c_S^i, c_U^i) = |\sigma^i - \sigma^{(i-1)}| / |\sigma^{(i-1)}| \quad (1)$$

3 Hyponymy Relation Acquisition

In this section we explain how each process extracts hyponymy relations from its respective text source either Wikipedia or Web pages. Each process extracts hyponymy relation candidates (denoted by (*hyper,hypo*) in this section). Because there are many non-hyponymy relations in these candidates[2], we classify hyponymy relation candidates into correct hyponymy relation or not. We used SVMs (Vapnik, 1995) for the classification in this paper.

3.1 Acquisition from Wikipedia



(a) Layout structure (b) Tree structure

Figure 3: Example borrowed from Oh et al. (2009): Layout and tree structures of Wikipedia article TIGER

We follow the method in Oh et al. (2009) for acquiring hyponymy relations from the Japanese Wikipedia. Every article is transformed into a tree structure as shown in Fig. 3, based on the items in its hierarchical layout including *title*, *(sub)section headings*, and *list items*. Candidate relations are extracted from this tree structure by regarding a node as a hypernym candidate and all of its subordinate nodes as potential hyponyms of the hypernym candidate (e.g., (TIGER, TAXONOMY) and (TIGER, SIBERIAN TIGER) from Fig. 3). We obtained 1.9×10^7 Japanese hyponymy relation candidates from Wikipedia.

[2]Only 25–30% of candidates was true hyponymy relation in our experiments.

	Type	Description
Feature from Wikipedia ("WikiFeature")	Lexical	Morphemes and POS of *hyper* and *hypo*; *hyper* and *hypo* themselves
	Structure	Distance between *hyper* and *hypo* in a tree structure; Lexical patterns for article or section names, where listed items often appear; Frequently used section headings in Wikipedia (e.g., "Reference"); Layout item type (e.g., section or list); Tree node type (e.g., root or leaf); Parent and children nodes of *hyper* and *hypo*
	Infobox	Attribute type and its value obtained from Wikipedia infoboxes
Feature from Web texts ("WebFeature")	Lexical	Morphemes and POS of *hyper* and *hypo*; *hyper* and *hypo* themselves
	Pattern	Lexico-syntactic patterns applied to *hyper* and *hypo*; PMI score between pattern and hyponymy relation candidate *(hyper,hypo)*
	Collocation	PMI score between *hyper* and *hypo*
	Noun Class	Noun classes relevant to *hyper* and *hypo*

Table 1: Feature sets (WikiFeature and WebFeature): *hyper* and *hypo* represent hypernym and hyponym parts of hyponymy relation candidates, respectively.

As features for classification we used lexical, structure, and infobox information from Wikipedia (WikiFeature), as shown in Table 1. Because they are the same feature sets as those used in Oh et al. (2009), here we just give a brief overview of the feature sets. Lexical features[3] are used to recognize the lexical evidence for hyponymy relations encoded in *hyper* and *hypo*. For example, the common head morpheme *tiger* in (TIGER, BENGAL TIGER) can be used as the lexical evidence. Such information is provided along with the words/morphemes and the parts of speech of *hyper* and *hypo*, which can be multi-word/morpheme nouns.

Structure features provide evidence found in layout or tree structures for hyponymy relations. For example, hyponymy relations (TIGER, BENGAL TIGER) and (TIGER,MALAYAN TIGER) can be obtained from tree structure "(root node, children nodes of *Subspecies*)" in Fig 3.

3.2 Acquisition from Web Texts

As the target for hyponymy relation acquisition from the Web, we used 5×10^7 pages from the TSUBAKI corpus (Shinzato et al., 2008), a 10^8 page Japanese Web corpus that was dependency parsed with KNP (Kurohashi-Nagao Parser) (Kurohashi and Kawahara, 2005). Hyponymy relation candidates are extracted from the corpus based on the lexico-syntactic patterns such as "*hypo* nado *hyper* (*hyper* such as *hypo*)" and "*hypo* to iu *hyper* (*hyper* called *hypo*)" (Ando et al., 2004). We extracted 6×10^6 Japanese hyponymy relation candidates from the Japanese Web texts. Features (WebFeature) used for classification are summarized in Table 1. Similar to the hyponymy relation acquisition from Wikipedia, lexical features are used to recognize the lexical evidence for hyponymy relations.

Lexico-syntactic patterns for hyponymy relation show different coverage and accuracy in hyponymy relation acquisition (Ando et al., 2004). Further if multiple lexico-syntactic patterns support acquisition of hyponymy relation candidates, these candidates are more likely to be actual hyponymy relations. The pattern feature of hyponymy relation candidates is used for these evidence.

We use PMI (point-wise mutual information) of hyponymy relation candidate (*hyper*, *hypo*) as a collocation feature (Pantel and Ravichandran, 2004), where we assume that *hyper* and *hypo* in candidates would frequently co-occur in the same sentence if they hold a hyponymy relation.

Semantic noun classes have been regarded as useful information in semantic relation acquisition (De Saeger et al., 2009). EM-based clustering (Kazama and Torisawa, 2008) is used for obtaining 500 semantic noun classes[4] from 5×10^5 nouns (including single-word and multi-word ones) and their 4×10^8 dependency relations with 5×10^5 verbs and other nouns in our target Web

[3]MeCab (`http://mecab.sourceforge.net/`) was used to provide the lexical features.

[4]Because EM clustering provides a probability distribution over noun class nc, we obtain discrete classes of each noun n with a probability threshold $p(nc|n) \geq 0.2$ (De Saeger et al., 2009).

	Co-training (Blum and Mitchell, 1998)	Bilingual co-training (Oh et al., 2009)	Co-STAR (Proposed method)
Instance space	Same	Different	Almost different
Feature space	Split by human decision	Split by languages	Split by source texts
Common instances	Genuine-common (or All unlabeled) instances	Genuine-common instances (Translatable)	Genuine-common and virtual-common instances

Table 2: Differences among co-training, bilingual co-training, and Co-STAR

corpus. For example, noun class C_{311} includes biological or chemical substances such as *tatou* (*polysaccharide*) and *yuukikagoubutsu* (*organic compounds*). Noun classes (i.e., C_{311}) relevant to *hyper* and *hypo*, respectively, are used as a noun class feature.

4 Related Work

There are two frameworks, which are most relevant to our work – bilingual co-training and ensemble semantics.

The main difference between bilingual co-training and Co-STAR lies in an instance space. In bilingual co-training, instances are in different spaces divided by languages while, in Co-STAR, many instances are in different spaces divided by their source texts. Table 2 shows differences between co-training, bilingual co-training and Co-STAR.

Ensemble semantics is a relation acquisition framework, where semantic relation candidates are extracted from multiple sources and a single ranker ranks or classifies the candidates in the final step (Pennacchiotti and Pantel, 2009). In ensemble semantics, one ranker is in charge of ranking all candidates extracted from multiple sources; while one classifier classifies candidates extracted from one source in Co-STAR.

5 Experiments

We used the July version of Japanese Wikipedia (jawiki-20090701) as structured text. We randomly selected 24,000 hyponymy relation candidates from those identified in Wikipedia and manually checked them. 20,000 of these samples were used as training data for our initial classifier, the rest was equally divided into development and test data for Wikipedia. They are called "WikiSet." As unstructured text, we used 5×10^7 Japanese Web pages in the TSUBAKI corpus (Shinzato et al., 2008). Here, we manually checked 9,500 hyponymy relation candidates selected randomly from Web texts. 7,500 of these were used as training data. The rest was split into development and test data. We named this data "WebSet".

In both classifiers, the development data was used to select the optimal parameters, and the test data was used to evaluate our system. We used TinySVM (TinySVM, 2002) with a polynomial kernel of degree 2 as a classifier. α (the threshold value indicating high confidence), β (the threshold value indicating low confidence), and $TopN$ (the maximum number of training instances to be added to the existing training data in each iteration) were selected through experiments on the development set. The combination of $\alpha = 1$, $\beta = 0.3$, and $TopN$=900 showed the best performance and was used in the following experiments. Evaluation was done by precision (P), recall (R), and F-measure (F).

5.1 Results

We compare six systems. Three of these, B1–B3, show the effect of different feature sets ("WikiFeature" and "WebFeature" in Table 1) and different training data. We trained two separate classifiers in B1 and B2, while we integrated feature sets and training data for training a single classifier in B3. The classifiers in these three systems are trained with manually prepared training data ("WikiSet" and "WebSet"). For the purpose of our experiment, we consider **B3** as the closest possible approximation of the ensemble semantics framework (Pennacchiotti and Pantel, 2009).

- **B1** consists of two completely independent classifiers. Both S and U classifiers are trained and tested on their own feature and data sets (respectively "WikiSet + WikiFeature" and "WebSet + WebFeature").

- **B2** is the same as B1, except that both classifiers are trained with all available training data — WikiSet and WebSet are combined (27,500 training instances in total). However, each classifier only uses its own feature set (WikiFeature or WebFeature)[5].

- **B3** adds a *master* classifier to **B1**. This third classifier is trained on the complete 27,500 training instances (same as **B2**) using all available features from Table 1, including each instance's SVM scores obtained from the two **B1** classifiers[6]. The verdict of the master classifier is considered to be the final classification result.

The other three systems, BICO, Co-B, and Co-STAR (our proposed method), are for comparison between bilingual co-training (Oh et al., 2009) (BICO) and variants of Co-STAR (Co-B and Co-STAR). Especially, we prepared Co-B and Co-STAR to show the effect of different configurations of common instances on the Co-STAR algorithm. We use both B1 and B2 as the initial classifiers of Co-B and Co-STAR. We notate Co-B and Co-STAR without '$*$' when B1 is used as their initial classifier and those with '$*$' when B2 is used.

- **BICO** implements the bilingual co-training algorithm of (Oh et al., 2009), in which two processes collaboratively acquire hyponymy relations in two *different languages*. For BICO, we prepared 20,000 English and 20,000 Japanese training samples (Japanese ones are the same as training data in the WikiSet) by hand.

- **Co-B** is a variant of Co-STAR that uses only the genuine-common instances as common instances (67,000 instances)[7], to demonstrate the effectiveness of the virtual common instances.

- **Co-STAR** is our proposed method, which uses both genuine-common and virtual-common instances (643,000 instances in total).

	WebSet			WikiSet		
	P	R	F	P	R	F
B1	84.3	65.2	73.5	87.8	74.7	80.7
B2	83.4	69.6	75.9	87.4	79.5	83.2
B3	82.2	72.0	76.8	86.1	77.7	81.7
BICO	N/A	N/A	N/A	84.5	**81.8**	83.1
Co-B	**86.2**	63.5	73.2	**89.7**	74.1	81.2
Co-B$*$	85.5	69.9	77.0	89.6	76.5	82.5
Co-STAR	85.9	76.0	80.6	88.0	**81.8**	**84.8**
Co-STAR$*$	83.3	**80.7**	**82.0**	87.6	**81.8**	84.6

Table 3: Comparison of different systems

Table 3 summarizes the result. Features for common instances in Co-B and Co-STAR are prepared in the same way as training data in B2, so that both classifiers can classify the common instances with their trained feature sets.

Comparison between B1–B3 shows that B2 and B3 outperform B1 in F-measure. More training data used in B2–B3 (27,500 instances for both WebSet and WikiSet) results in higher performance than that of B1 (7,500 and 20,000 instances used separately). We think that the lexical features, assigned regardless of source text to instances in B2–B3, are mainly responsible for the performance gain over B1, as they are the least domain-dependent type of features. B2–B3 are composed of different number of classifiers, each of which is trained with different feature sets and training instances. Despite this difference, B2 and B3 showed similar performance in F-measure.

Co-STAR outperformed the algorithm similar to the ensemble semantics framework (B3), although we admit that a more extensive comparison is desirable. Further Co-STAR outperformed BICO. While the manual cost for building the initial training data used in Co-STAR and BICO is hard to quantify, Co-STAR achieves better performance with fewer training data in total (27,500 instances) than BICO (40,000 instances). The difference in performance between Co-B and Co-STAR shows the effectiveness of

[5]Note that training instances from WebSet (or WikiSet) can have WikiFeature (or WebFeature) if they also appear in Wikipedia (or Web corpus). But they can always have lexical feature, the common feature set between WikiFeature and WebFeature.

[6]SVM scores are assigned to the instances in training data in a 10-fold cross validation manner.

[7]Co-B can be considered as conventional co-training (Blum and Mitchell, 1998) in the sense that two classifiers collaborate through actual common instances.

the automatically generated virtual-common instances. From these comparison, we can see that virtual-common instances coupled with genuine-common instances can be leveraged to enable more effective collaboration between the two classifiers in Co-STAR.

As a result, our proposed method outperforms the others in F-measure by 1.4–8.5%. We obtained 4.3×10^5 hyponymy relations from Web texts and 4.6×10^6 ones from Wikipedia[8].

6 Co-STAR with Automatically Generated Training Data

For Co-STAR, we need two sets of manually prepared training data, one for structured text and the other for unstructured text. As in any other supervised system, the cost of preparing the training data is an important issue. We therefore investigated whether Co-STAR can be trained for a lower cost by generating more of its training data automatically.

We automatically built training data for Web texts by using definition sentences[9] and category names in the Wikipedia articles, while we stuck to manually prepared training data for Wikipedia. To obtain hypernyms from Wikipedia article names, we used definition-specific lexico-syntactic patterns such as "*hyponym* is *hypernym*" and "*hyponym* is a type of *hypernym*" (Kazama and Torisawa, 2007; Sumida and Torisawa, 2008). Then, we extracted hyponymy relations consisting of pairs of Wikipedia category names and their member articles when the Wikipedia category name and the hypernym obtained from the definition of the Wikipedia article shared the same head word. Next, we selected a subset of the extracted hyponymy relations that are also hyponymy relation candidates in Web texts, as positive instances for hyponymy relation acquisition from Web text. We obtained around 15,000 positive instances in this way. Negative instances were chosen from virtual-common instances, which also originated from the Wikipedia category system and hyponymy relation candidates in Web texts (around 293,000 instances).

The automatically built training data was noisy and its size was much bigger than manually prepared training data in WebSet. Thus 7,500 instances as training data (the same number of manually built training data in WebSet) were randomly chosen from the positive and negative instances with a positive:negative ratio of 1:4[10].

	WebSet			WikiSet		
	P	R	F	P	R	F
B1	81.0	47.6	60.0	**87.8**	74.7	80.7
B2	80.0	55.4	65.5	87.1	79.5	83.1
B3	82.0	33.7	47.8	87.1	75.6	81.0
Co-STAR	**82.2**	60.8	69.9	87.3	80.7	83.8
Co-STAR$*$	79.2	**69.6**	**74.1**	87.0	**81.8**	**84.4**

Table 4: Results with automatically generated training data

With the automatically built training data for Web texts and manually prepared training data for Wikipedia, we evaluated B1–B3 and Co-STAR, which are the same systems in Table 3. The results in Table 4 are encouraging. Co-STAR was robust even when faced with noisy training data. Further Co-STAR showed better performance than B1–B3, although its performance in Table 4 dropped a bit compared to Table 3. This result shows that we can reduce the cost of manually preparing training data for Co-STAR with only small loss of the performance.

7 Conclusion

This paper proposed Co-STAR, an algorithm for hyponymy relation acquisition from structured and unstructured text. In Co-STAR, two independent processes of hyponymy relation acquisition from structured texts and unstructured texts, collaborate in an iterative manner through common instances. To improve this collaboration, we introduced virtual-common instances.

Through a series of experiments, we showed that Co-STAR outperforms baseline systems and virtual-common instances can be leveraged to achieve better performance. We also showed that Co-STAR is robust against noisy training data, which requires less human effort to prepare it.

[8] We obtained them with 90% precision by setting the SVM score threshold to 0.23 for Web texts and 0.1 for Wikipedia.

[9] The first sentences of Wikipedia articles.

[10] We select the ratio by testing different ratio from 1:2 to 1:5 with our development data in WebSet and B1.

References

Ando, Maya, Satoshi Sekine, and Shun Ishiza. 2004. Automatic extraction of hyponyms from Japanese newspaper using lexico-syntactic patterns. In *Proc. of LREC '04*.

Blum, Avrim and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *COLT' 98: Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100.

De Saeger, Stijn, Kentaro Torisawa, Jun'ichi Kazama, Kow Kuroda, and Masaki Murata. 2009. Large scale relation acquisition using class dependent patterns. In *Proc. of ICDM 2009*, pages 764–769.

Hearst, Marti A. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545.

Kazama, Jun'ichi and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proc. of Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 698–707.

Kazama, Jun'ichi and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of ACL-08: HLT*, pages 407–415.

Kurohashi, Sadao and Daisuke Kawahara. 2005. KNP (Kurohashi-Nagao Parser) 2.0 users manual.

Oh, Jong-Hoon, Kiyotaka Uchimoto, and Kentaro Torisawa. 2009. Bilingual co-training for monolingual hyponymy-relation acquisition. In *Proc. of ACL-09: IJCNLP*, pages 432–440.

Pantel, Patrick and Deepak Ravichandran. 2004. Automatically labeling semantic classes. In *Proc. of HLT-NAACL '04*, pages 321–328.

Pantel, Patrick, Eric Crestan, Arkady Borkovsky, Ana-Maria Popescu, and Vishnu Vyas. 2009. Web-scale distributional similarity and entity set expansion. In *Proceedings of EMNLP '09*, pages 938–947.

Pennacchiotti, Marco and Patrick Pantel. 2009. Entity extraction via ensemble semantics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 238–247.

Ravi, Sujith and Marius Pasca. 2008. Using structured text for large-scale attribute extraction. In *CIKM-08*, pages 1183–1192.

Shinzato, Keiji and Kentaro Torisawa. 2004. Extracting hyponyms of prespecified hypernyms from itemizations and headings in web documents. In *Proceedings of COLING '04*, pages 938–944.

Shinzato, Keiji, Tomohide Shibata, Daisuke Kawahara, Chikara Hashimoto, and Sadao Kurohashi. 2008. Tsubaki: An open search engine infrastructure for developing new information access. In *Proceedings of IJCNLP '08*, pages 189–196.

Snow, Rion, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 801–808.

Suchanek, Fabian M., Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proc. of WWW '07*, pages 697–706.

Sumida, Asuka and Kentaro Torisawa. 2008. Hacking Wikipedia for hyponymy relation acquisition. In *Proc. of the Third International Joint Conference on Natural Language Processing (IJCNLP)*, pages 883–888, January.

Talukdar, Partha Pratim, Joseph Reisinger, Marius Pasca, Deepak Ravichandran, Rahul Bhagat, and Fernando Pereira. 2008. Weakly-supervised acquisition of labeled class instances using graph random walks. In *Proc. of EMNLP08*, pages 582–590.

TinySVM. 2002. `http://chasen.org/~taku/software/TinySVM`.

Van Durme, Benjamin and Marius Pasca. 2008. Finding cars, goddesses and enzymes: Parametrizable acquisition of labeled instances for open-domain information extraction. In *Proc. of AAAI08*, pages 1243–1248.

Vapnik, Vladimir N. 1995. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA.

Wang, Wei and Zhi-Hua Zhou. 2007. Analyzing co-training style algorithms. In *ECML '07: Proceedings of the 18th European conference on Machine Learning*, pages 454–465.