Heuristic Methods for Reducing Errors of Geographic Named Entities Learned by Bootstrapping

Seungwoo Lee and Gary Geunbae Lee

Department of Computer Science and Engineering,
Pohang University of Science and Technology,
San 31, Hyoja-dong, Nam-gu, Pohang, 790-784, Republic of Korea

Abstract. One of issues in the bootstrapping for named entity recognition is how to control annotation errors introduced at every iteration. In this paper, we present several heuristics for reducing such errors using external resources such as WordNet, encyclopedia and Web documents. The bootstrapping is applied for identifying and classifying fine-grained geographic named entities, which are useful for applications such as information extraction and question answering, as well as standard named entities such as PERSON and ORGANIZATION. The experiments show the usefulness of the suggested heuristics and the learning curve evaluated at each bootstrapping loop. When our approach was applied to a newspaper corpus, it could achieve 87 F1 value, which is quite promising for the fine-grained named entity recognition task.

1 Introduction

A bootstrapping process for named entity recognition is usually as follows. In the initial stage, it selects seeds and annotates a raw corpus using the seeds. From the annotation, internal and contextual patterns are learned and applied to the corpus again to obtain new candidates of each type. Several methods are adopted to reduce over-generation and incorrect annotation and accept only correct ones. *One sense per discourse* heuristic may also be adopted to expand the annotated instances. It repeats until no more new patterns and entities are learned.

There are several issues in bootstrapping approaches for named entity recognition task to achieve successful performance. One of them is how to control annotation errors introduced in the bootstrapping process, on which we are focusing in this paper. As iteration continues, the bootstrapping expands previous annotation to increase recall. But this expansion may also introduce annotation errors and, as a result, decrease the precision. Ambiguous entities may be misclassified since learning speed per class depends on seeds. For example, *'New York'* may be misclassified to a city name before the patterns that correctly classify it to a state name are learned. Especially such errors in the early stage of the bootstrapping are quite harmful because the errors are accumulated. The

R. Dale et al. (Eds.): IJCNLP 2005, LNAI 3651, pp. 658–669, 2005.

annotation errors are classified into following four cases: *inclusion*, *crossing*, *type conflict* and *spurious*. The first three errors occur when a learned entity overlaps a true entity whereas the last one occurs when a learned entity does not overlap any true entity. Most previous works depend only on the statistics (e.g., scores of patterns) obtained from the previous annotation to control such errors. However, this strategy is not always the best because some trivial errors can also be corrected by simple heuristics. We suggest several heuristics that control the annotation errors in Section 4. The heuristics are embedded in a bootstrapping algorithm, which is modified and improved from [4] and shortly described in Section 3.

Unlike the traditional named entity task, we deal with sub-categorized geographic named entities (i.e., locations) in addition to PERSON and ORGANIZATION. Geographic named entities can be classified into many sub-types that are critical for applications such as information extraction and question answering. As a first step, we define their ten sub-classes: COUNTRY, STATE, COUNTY, CITY, MOUNTAIN, RIVER, ISLAND, LAKE, CONTINENT and OCEAN. We attempt to identify and classify all instances of the eleven classes as well as PERSON and ORGANIZATION in plain text. Annotation of geographic named entities is a formidable task. Geographic named entities are frequently shared between their sub-classes as well as with person names. For example, *'Washington'* may indicate a person in one context but may also mean a city or state in another context. Even country names cannot be exceptions. For some Americans, *'China'* and *'Canada'* may be cities where they live. Geographic named entities such as *'Turkey'* and *'Chile'* can also be shared with common nouns. Contextual similarity among geographic named entities is much higher than the one between PLO (Person-Location-Organization) entities since they are much closer semantically. These make geographic named entity annotation task more difficult than that of the traditional named entity task.

The remainder of this paper is as follows. Section 2 presents and compares related works to our approach. The bootstrapping algorithm is shortly described in Section 3 and several heuristics for controlling the annotation errors are explained in Section 4. Section 5 gives some experimental results verifying our approach, which is followed by conclusions and future works in Section 6.

2 Related Works

Most bootstrapping approaches start with incomplete annotations and patterns obtained from selected seeds and learn to obtain more complete annotations and patterns. However, the incompleteness is apt to cause annotation errors to be introduced in each bootstrapping iteration. Most previous works have designed their own statistical measures to control such errors. Phillips and Riloff [9] developed *evidence* and *exclusivity* measures to filter out ambiguous terms and Yangarber et al. [12] calculated *accuracy*, *confidence* and *score* of their patterns to select better patterns. However, those statistical measures are calculated only using data obtained from their training corpus which cannot often

give enough information. Instead, other resources like World Wide Web as well as a gazetteer can be incorporated to compensate the lack of information from the training corpus.

Research on analysis of geographic references recently started to appear and has two directions. One is to focus on building gazetteer databases [6,11] and the other is to focus on classifying geographic entity instances in text [5].

Manov et al. [6] presented KIM (Knowledge and Information Management) that consists of an ontology and a knowledge base. They used it for information extraction but did not show notable results. Uryupina [11] presented a bootstrapping method to obtain gazetteers from the internet. By searching for seed names on the internet, she obtained lexical patterns and learned each classifier for six location sub-types, such as COUNTRY, CITY, ISLAND, RIVER, MOUNTAIN and REGION. Then she obtained and classified candidate names by searching the patterns in the internet. Li et al. [5] suggested a hybrid approach to classify geographic entities already identified as location by an existing named entity tagger. They first matched local context patterns and then used a maximum spanning tree search for discourse analysis. They also applied a default sense heuristic as well as one sense per discourse principle. According to their experiments, the default sense heuristic showed the highest contribution.

3 Bootstrapping

Our bootstrapping algorithm was modified and improved from [4] and the bootstrapping flow has one initial step and four iterative steps, as shown in Figure 1. In the initial step, we annotate a raw corpus with seeds automatically obtained from various gazetteers. Starting and ending boundary patterns are learned from the annotation and applied to the corpus again to obtain new candidates of each type. Then we eliminate annotation errors in the candidates using several

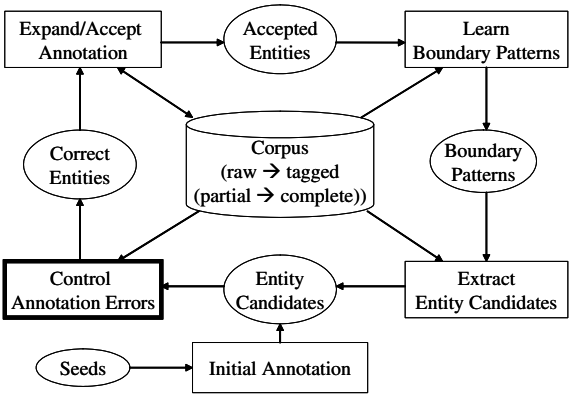

Fig. 1. The bootstrapping overview

linguistic heuristics, which is described in detail in Section 4. Finally, the remaining entity candidates propagate their annotations into other occurrences within the same document by *one sense per discourse* principle [2]. This loop continues until there are no new patterns learned. The algorithm is summarized as follows:

Step 0: Seed Preparation and Initial Annotation

We prepare seeds from the gazetteer and obtain initial entity candidate set, C_1, by marking occurrences of the seeds in the training raw corpus.

$C_1 = \{e_i | e_i$ is an entity candidate obtained from seeds but not accepted yet$\}$;

And we initialize the number of iteration (k), the set of accepted boundary patterns (P_0) and the set of accepted entities (E_0) as follows:

$k = 1$; $P_0 = \phi$; $E_0 = \phi$;

Step 1: Controlling the Annotation Errors

We filter out annotation errors among the entity candidates (C_k) using several heuristics with external resources and construct E_k, a set of entities checked as correct (see Section 4).

$E_k = \{e_i | e_i \in C_k$ and e_i is checked as correct by heuristics$\}$;

Step 2: Expanding and Accepting the Annotation

After removing erroneous candidates, we expand the correct entities by applying *one sense per document* heuristic and then accept M[1] top-ranked entities to construct a new E_k, the set of currently accepted entities.

$E_k = \{e_i | e_i \in E_k$ or is an instance expanded from $e_j \in E_k$, and $Rank(e_i) \geq Rank(e_{i+1}), 1 \leq i \leq M\}$;
$E_k = E_{k-1} \cup E_k$;

The rank of an entity candidate, $Rank(e_i)$, is computed as follows:

$$Rank(e_i) = 1 - \{1 - Score(BP_s(e_i))\} \times \{1 - Score(BP_e(e_i))\} \quad (1)$$

$BP_s(e)$ and $BP_e(e)$ indicate starting and ending boundary patterns of an entity e, respectively.

[1] M was set to 300 in our experiment.

Step 3: Learning Boundary Patterns

From the currently accepted entity set, E_k, we learn a new boundary pattern candidate set, $\tilde{P}_k$. We generate starting and ending boundary patterns and compute the accuracy ($Acc(p_i)$) of each pattern p_i which is used to filter out inaccurate patterns below Θ_a[2] and construct $\tilde{P}_k$. Then we compute the score ($Score(p_i)$) of each pattern p_i and add new N[3] top-scored patterns among $\tilde{P}_k$ to the accepted boundary pattern set, P_k, if there exist new patterns in $\tilde{P}_k$. Otherwise, the bootstrapping process stops.

$\tilde{P}_k = \{p_i | p_i = BP(e), e \in E_k$ and $p_i \notin P_{k-1}$ and $Acc(p_i) \geq \Theta_a\}$;
If $\tilde{P}_k = \phi$ then stop;
Otherwise, $P_k = P_{k-1} \cup \{p_i | p_i \in \tilde{P}_k$ and $Score(p_i) \geq Score(p_{i+1}), 1 \leq i \leq N\}$;

The accuracy, $Acc(p)$ and the score, $Score(p)$, of a boundary pattern, p, are computed as follows:

$$Acc(p) = \frac{pos(p)}{pos(p) + neg(p)} \times \frac{1 - \frac{1}{pos(p)^2+1}}{1 - \frac{1}{Np^2+1}}, \tag{2}$$

$$Score(p) = \frac{pos(p)}{pos(p) + 2 \times neg(p) + unk(p)} \times \frac{1 - \frac{1}{\ln(pos(p)+3)}}{1 - \frac{1}{\ln(Np+3)}}, \tag{3}$$

where $pos(p)$ is the number of instances that are matched to p and already annotated with the same entity type; $neg(p)$ is the number of instances that are matched to p but already annotated with a different type or previously filtered out; $unk(p)$ is the number of instances that are matched to p but not annotated yet; Np is the maximum value of $pos(p)$.

Step 4: Applying Boundary Patterns and Extracting Candidates

We extract new entity candidates, C_{k+1}, for the next iteration by applying the accepted boundary patterns, P_k, to the training corpus and then go to Step 1.

$C_{k+1} = \{e_i | BP_s(e_i) \in P_k$ and $BP_e(e_i) \in P_k$ and $e_i \notin E_k\}$;
$k := k + 1$;
Go to Step 1.

Since each pattern determines only one – i.e., *starting* or *ending* – boundary, a candidate is identified and classified by a pair of starting and ending boundary patterns with the same type.

[2] Θ_a was set to 0.1 in our experiment.
[3] N was set to 700 in our experiment.

4 Error Controls

The annotation errors introduced in the bootstrapping process are classified into following four cases, based on the inconsistency between an erroneous entity candidate and a true entity: *inclusion*, *crossing*, *type conflict* and *spurious*. *Inclusion* occurs when a candidate is a sub-phrase of a true entity – e.g., *'U.S.'* in *'U.S. Army'*. *Crossing* occurs when a candidate partially overlaps with a true entity – e.g., *'Columbia River'* in *"British Columbia River"*, which means a river in *'British Columbia'*. *Type conflict* occurs when a candidate has the same text span but different type from a true entity – e.g., *'New York'* may be misclassified into STATE but it is CITY. *Spurious* indicates that a candidate is spurious and does not interfere with any true entities.

To resolve these inconsistencies, we basically use statistical measures such as the score of a boundary pattern, $Score(p)$, and the rank of an entity candidate, $Rank(e)$, as in most previous works. However, this strategy is not always the best because some trivial errors can also be removed by simple heuristics and linguistic knowledge. Especially, the strategy cannot be applied to erroneous entities whose inconsistencies cannot be detected since their true entities are not identified yet. We call it *potential* inconsistency. We examine *potential inclusion* and *potential type conflict* for each entity candidate using the gazetteer and Web resources. To overcome this limitation of statistical measures obtained from the training corpus, we design several methods that incorporate linguistic knowledge and external resources, which are described in the following subsections.

4.1 Co-occurrence Information

Co-occurrence information (CI) has been widely used to resolve word sense ambiguity [3,8,10] and also can be employed to resolve *crossing* and *type conflict* inconsistencies, which can be regarded as word sense ambiguity problem. We assume that two instances of an ambiguous entity that occur in different texts can be classified into the same class if they share their CI. CI can be collected from definition statements of an entity of an encyclopedia. For example, the underlined phrases are collected as CI of an entity *'Clinton'* with class CITY from a statement *"Clinton is a city in Big Stone County, Minnesota, USA"*. In this way, we could construct initial CI for 18000 entities from the Probert Encyclopedia (http://www.probertencyclopaedia.com/places.htm), most of which are geographic entities. We also augment CI from the accepted entity instances during the bootstrapping process. We consider capitalized nouns or noun phrases in the window of up to left/right 60 words, within sentence boundary, from an entity as its CI. Then, the score of an entity e with class t, $Coinfo(e,t)$, is calculated as the similarity of CI:

$$Coinfo(e,t) = \frac{\sum_{i=1}^{N} freq(cw_i, e, t) \times count(cw_i, e)}{N}, \tag{4}$$

where N is the number of co-occurrence information cw_i, $freq(cw_i, e, t)$ means the frequency of cw_i co-occurring with an entity e of class t in the learned

co-occurrence information and $count(cw_i, e)$ means the frequency of cw_i co-occurring with the entity in the current pending context. When two candidates cause *crossing* or *type conflict*, the candidate having smaller $Coinfo$ is considered to be incorrect and removed.

4.2 Gazetteer with Locator

Most entities are often mentioned with geographic entities where they are located, especially when they are not familiar to general readers. For example, *'Dayton'* in *"the Dayton Daily News, Dayton, Ohio"* is restricted to an entity in *'Ohio'*. This means that we can classify *'Dayton'* into CITY if we know a fact that there is a city named *'Dayton'* and located at *'Ohio'*. We can say that the locator information is a special case of the co-occurrence information. The locator information was also collected from the Probert Encyclopedia. If one of two entity candidates causing *crossing* or *type conflict* has a verified locator, the other can be regarded as an error and removed.

4.3 Prior Probability

Ambiguous entities often have different prior probability according to each class. For example, *'China'* appears frequently in general text as a country name but rarely as a city name. *'Canada'* is another example. This means that when two entity candidates cause *type conflict* we can remove one having lower prior probability. It is hard to acquire such probabilities if we do not have a large annotated corpus. However, WordNet [7] can give us the information that is needed to infer the relative prior probability since the sense order in WordNet reflects the frequency that the sense appears in text. According to WordNet, for example, *'New York'* is more frequently mentioned as a city name than as a state name and, therefore, is classified into CITY if its context does not give strong information that it is a state name. We could construct relative prior probabilities for 961 ambiguous gazetteer entries from WordNet. The prior probability of entity e with type t based on WordNet, $Prior_{WN}(e,t)$, is calculated as follows:

$$Prior_{WN}(e,t)=\begin{cases} \frac{1}{N+1} + \alpha_{WN} \times \frac{(m+1)-Sense\#_{WN}(e,t)}{\sum_{i=1}^{m} i} & \text{if there exist in WordNet} \\ \frac{1}{N+1} - \beta_{WN} & \text{otherwise,} \end{cases} \tag{5}$$

where N is the number of possible types of entity candidate e, m is the number of types of entity candidate e registered in WordNet, and $Sense\#_{WN}(e,t)$ means the WordNet sense no. of entity candidate e with type t. α_{WN} and β_{WN} are calculated as follows:

$$\alpha_{WN} = \beta_{WN} \times (N-m) + \frac{1}{N+1}$$

$$\beta_{WN} = \frac{m}{(N+1)^2}$$

Based on these formulas, the prior probabilities of an entity '*New York*' are given as follows according to its type: (CITY, 0.44), (STATE, 0.32), (TOWN, 0.12), and (COUNTY, 0.12).[4]

Although this prior probability is quite accurate, it does not have sufficient applicability. Therefore, we need to develop another method that can acquire prior probabilities of much more entities and Web can be one alternative. For each ambiguous entity X, we query *"X is a/an"* to at least two Web search engines[5] and extract and collect a noun phrase Y matching to *"X is a/an Y"*. Then, we determine a type, which Y belongs to, using WordNet and count its frequency. This frequency for each possible type of the entity X is regarded as sense order information. That is, we can assign to each possible type a sense number in the descending order of the frequency. Now, the prior probability of an entity e with type t based on the Web, $Prior_{Web}(e,t)$, can be similarly calculated. Then, the final prior probability, $Prior(e,t)$, is computed by arithmetic mean of $Prior_{WN}(e,t)$ and $Prior_{Web}(e,t)$. Combined with the Web search, the prior probabilities of the above example are changed as follows: (CITY, 0.36), (STATE, 0.29), (TOWN, 0.18), and (COUNTY, 0.17).

4.4 Default Type

When an ambiguous candidate causing *type conflict* is not registered in WordNet and cannot be detected by the Web search, we can apply default type heuristic. Unlike the prior probability, default type indicates a priority between any two target classes regardless of each individual entity. In general, we can say that, for an ambiguous entity between COUNTRY and CITY, COUNTRY is more dominant than CITY since a country name is more familiar to common people. We built up default types between all pairs of target classes using human linguistic knowledge and prior probability described in the previous subsection.

4.5 Part of Other Entity

Potential inclusion is often not exposed at a bootstrapping iteration since boundary patterns for each class are generated at different speeds and, in addition, all required boundary patterns cannot be generated from seeds. For this, we design two methods in addition to gazetteer consulting.

First, we check if there exists an acronym for a super-phrase. [1] says that we can consult a commonly-used *acronym* to determine extent of a named entity. In other words, *"University of California, Los Angeles"*, for example, must

[4] WordNet does not have COUNTY and TOWN senses of '*New York*'.

[5] We used eight well-known Web search engines such as Google (http://www.google.com/), Ask Jeeves (http://web.ask.com/), AltaVista (http://www.altavista.com/), LookSmart (http://search.looksmart.com/), Teoma (http://s.teoma.com/), AlltheWeb (http://www.alltheweb.com/), Lycos (http://search.lycos.com/), and Yahoo! (http://search.yahoo.com/). We specially thank to the service providers.

be annotated as a unique organization name since the university is commonly referred to as `'UCLA'`. As an another example, *'U.S.'* in *"U.S. Navy"* should not be annotated as a country name but *'U.S.'* in *"U.S. President"* should be since *"U.S. Navy"* is represented as the acronym `'USN'` but *"U.S. President"* is not represented as `'USP'`. To check the existence of their acronyms, we can consult Web search engines by querying the suspected phrases with their possible acronyms, such as *"U.S. Navy* `(USN)` *"* and *"U.S. President* `(USP)` *"*, respectively, with exact match option.

Another solution is to check if a super-phrase beginning with a candidate whose class is one of geographic classes can be modified by a prepositional phrase which is derived by *in* or *comma (,)* plus the candidate (denoted as *in-loc*). For example, we can decide that *'Beijing'* in *"Beijing University"* is a part of the university name, since the phrase *"Beijing University* `in Beijing` *"* is found by Web search engines. If the *'Beijing'* denotes CITY, *"Beijing University"* means *a university* `in Beijing` and is not modified by the prepositional phrase "`in Beijing`" duplicately.

5 Experiments

The bootstrapping algorithm was developed and trained on part of New York Times articles (the first half of June, 1998; 28MB; 5,330 articles) from the AQUAINT corpus. We manually annotated 107 articles for test and the counts of annotated instances were listed in Table 1. A gazetteer composed of 80,000 entries was compiled from several Web sites[6]. This includes non-target entities as well as various aliases of entity names.

Table 1. The counts of instances annotated in the test corpus

COUNTRY	STATE	COUNTY	CITY	RIVER	MOUNT.	ISLAND	CONTI.	OCEAN	LAKE	PERSON	ORGAN.	Total
596	422	61	868	26	15	29	74	19	9	2,660	1,436	6,215

We first examined the usefulness of the heuristics, based on the instances (i.e., key instances) annotated in the test corpus. Applicability (*app.*) is defined as the number of key instances (denoted as *#app*), to which the heuristic can be applied, divided by the number of ambiguous ones (denoted as *#ambi*). Accuracy (*acc.*) is defined as the number of instances correctly resolved (denoted as *#corr*) divided by *#app*. There were 2250 ambiguous key instances in the test corpus.

[6] http://www.census.gov/, http://crl.nmsu.edu/Resources/resource.htm, http://www.timeanddate.com/, http://www.probertencyclopaedia.com/places.htm, http://www.world-of-islands.com/, and http://islands.unep.ch/isldir.htm

Applicability and accuracy of the first four heuristics for resolving *type conflict* are summarized in Table 2. As shown in the table, the first two heuristics – *co-occurrence information* and *gazetteer with locator* – have very low applicability but very high accuracy. On the contrary, the last two heuristics – *prior probability* and *default type* – show moderate accuracy with relatively high applicability. Based on this result, we combine the four heuristics in sequence such as high accurate one first and high applicable one last.

We also examined how well the heuristics such as *acronym* and *in-loc* can detect *potential inclusion* of an entity. In case of *acronym*, there were 2,555 key instances (denoted as *#app*) composed of more than one word and we searched the Web to check the existence of any possible acronym of each instance. As a result, we found out the correct acronyms for 1,143 instances (denoted as *#corr*). On the contrary, just 47 instances were incorrectly matched when we tried to search any acronyms of super-phrases of each key instance. In other words, *acronym* can detect *potential inclusion* at 46.58 applicability and 96.05 accuracy. In case of *in-loc*, 1,282 key instances beginning with a geographic word are tried to be checked if they appear with *in-loc* pattern in Web documents and 313 instances of them were confirmed. On the contrary, only 1 super-phrase of a key instance was incorrectly detected. Therefore, *in-loc* can detect *potential inclusion* at 24.49 applicability and 99.68 accuracy. These are summarized in Table 3. It says that the heuristics can detect quite accurately the extent of named entities although they do not have high applicability.

Finally, we evaluated the bootstrapping with the heuristics by investigating the performance change at every iteration. 50,349 seeds were selected from the gazetteer after removing ambiguous ones and only 3,364 seeds among them, which could be applied to the training corpus, were used for training. The recall and precision were measured using the standard MUC named entity scoring scheme and plotted in Figure 2. Starting at low recall and high precision, it gradually increases recall but slightly degrades precision, and it arrived at 87 F1

Table 2. Applicability and accuracy of the heuristics for resolving the inconsistency of 2,250 ambiguous instances (*#ambi*=2,250)

	#app	*#corr*	*app.*	*acc.*
co. info.	44	42	1.96	95.45
gaz. loc.	148	141	6.58	95.27
prior prob.	2,072	1,741	92.09	84.03
def. type	2,225	1,367	98.89	61.44

Table 3. Applicability and accuracy of the heuristics for detecting *potential inclusion*

	#ambi	*#app*	*#corr*	*app.*	*acc.*
acronym	2,555	1,190	1,143	46.58	96.05
in-loc	1,282	314	313	24.49	99.68

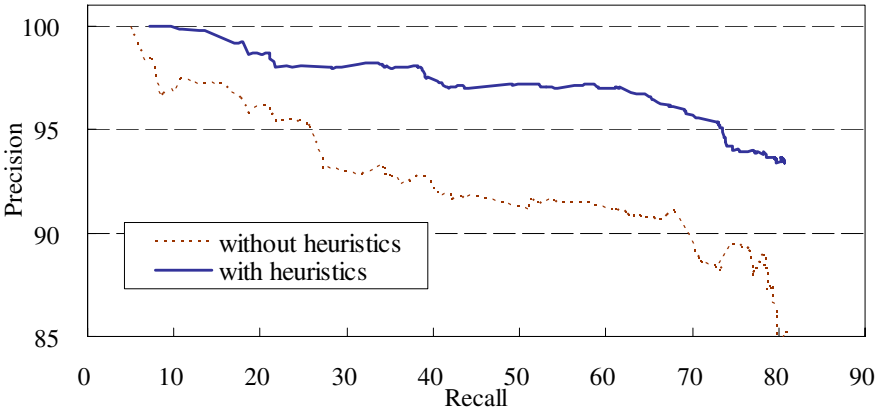

Fig. 2. The learning curve of the bootstrapping with the heuristics

(81 recall and 93 precision) after 1,100 iterations. We think that this performance is quite notable considering our fine-grained target classes, and the suggested heuristics work well to prevent incorrect entity candidates from being accepted during bootstrapping process.

6 Conclusions

In this paper, we observed four kinds of inconsistencies that degrade the performance of bootstrapping for named entity recognition with fine-grained geographic classes. To resolve such inconsistencies, we suggested several heuristics incorporating human linguistic knowledge and external resources like encyclopedia and Web documents. By analyzing the capability of each heuristic, we combined them in sequence. The bootstrapping with the heuristics was evaluated. Starting at low recall and high precision, the bootstrapping largely increased recall at a small cost of precision, and finally it achieved 87 F1. This means that the suggested approach is quite promising for the fine-grained named entity recognition task and the suggested heuristics can effectively reduce incorrect candidates introduced at the intermediate bootstrapping steps. In future, we plan to design a uniform statistical method that can augment the suggested heuristics especially using Web resources and also incorporate our heuristic knowledge used for filtering into the statistical model.

Acknowledgements

This work was supported by 21C Frontier Project on Human-Robot Interface (MOCIE) and by BK21 Project (Ministry of Education).

References

1. Chinchor, N., Brown, E., Ferro, L., Robinson, P.: 1999 Named Entity Recognition Task Definition (version 1.4). http://www.nist.gov/speech/tests/ie-er/er_99/doc/ne99_taskdef_v1_4.pdf (1999)
2. Gale, W.A., Church, K.W., Yarowsky, D.: One Sense Per Discourse. In: Proceedings of the 4th DARPA Speech and Natural Language Workshop. (1992) 233–237
3. Guthrie, J.A., Guthrie, L., Wilks, Y., Aidinejad, H.: Subject-dependent Co-occurrence and Word Sense Disambiguation. In: Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics (ACL), Berkeley, CA (1991) 146–152
4. Lee, S., Lee, G.G.: A Bootstrapping Approach for Geographic Named Entity Annotation. In: Proceedings of the 2004 Conference on Asia Information Retrieval Symposium (AIRS2004), Beijing, China (2004) 128–133
5. Li, H., Srihari, R.K., Niu, C., Li, W.: InfoXtract location normalization: a hybrid approach to geographic references in information extraction. In: Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References, Alberta, Canada (2003) 39–44
6. Manov, D., Kirjakov, A., Popov, B., Bontcheva, K., Maynard, D., Cunningham, H.: Experiments with geographic knowledge for information extraction. In: Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References, Alberta, Canada (2003) 1–9
7. Miller, G.A.: WordNet: A lexical database for English. Communications of the ACM **38** (1995) 39–41
8. Niwa, Y., Nitta, Y.: Co-occurrence Vectors from Corpora vs Distance Vectors from Dictionaries. In: Proceedings of the 15th International Conference on Computational Linguistics (COLING'94), Kyoto, Japan (1994) 304–309
9. Phillips, W., Riloff, E.: Exploiting Strong Syntactic Heuristics and Co-Training to Learn Semantic Lexicons. In: Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP2002), Philadelphia, PA (2002) 125–132
10. Shin, S., soek Choi, Y., Choi, K.S.: Word Sense Disambiguation Using Vectors of Co-occurrence Information. In: Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium (NLPRS2001), Tokyo, Japan (2001) 49–55
11. Uryupina, O.: Semi-supervised learning of geographical gazetteers from the internet. In: Proceedings of the HLT-NAACL 2003 Workshop on Analysis of Geographic References, Alberta, Canada (2003) 18–25
12. Yangarber, R., Lin, W., Grishman, R.: Unsupervised Learning of Generalized Names. In: Proceedings of the 19th International Conference on Computational Linguistics (COLING 2002), Taipei, Taiwan (2002) 1135–1141