A Bootstrapping Approach to Named Entity Classification Using Successive Learners

Cheng Niu, Wei Li, Jihong Ding, Rohini K. Srihari
Cymfony Inc.
600 Essjay Road, Williamsville, NY 14221. USA.
{cniu, wei, jding, rohini}@cymfony.com

Abstract

This paper presents a new bootstrapping approach to named entity (NE) classification. This approach only requires a few common noun/pronoun seeds that correspond to the concept for the target NE type, e.g. *he/she/man/woman* for PERSON NE. The entire bootstrapping procedure is implemented as training two successive learners: (i) a decision list is used to learn the parsing-based high precision NE rules; (ii) a Hidden Markov Model is then trained to learn string sequence-based NE patterns. The second learner uses the training corpus automatically tagged by the first learner. The resulting NE system approaches supervised NE performance for some NE types. The system also demonstrates intuitive support for tagging user-defined NE types. The differences of this approach from the co-training-based NE bootstrapping are also discussed.

1 Introduction

Named Entity (NE) tagging is a fundamental task for natural language processing and information extraction. An NE tagger recognizes and classifies text chunks that represent various proper names, time, or numerical expressions. Seven types of named entities are defined in the Message Understanding Conference (MUC) standards, namely, PERSON (PER), ORGANIZATION (ORG), LOCATION (LOC), TIME, DATE, MONEY, and PERCENT[1] (MUC-7 1998).

There is considerable research on NE tagging using different techniques. These include systems based on handcrafted rules (Krupka 1998), as well as systems using supervised machine learning, such as the Hidden Markov Model (HMM) (Bikel 1997) and the Maximum Entropy Model (Borthwick 1998).

The state-of-the-art rule-based systems and supervised learning systems can reach near-human performance for NE tagging in a targeted domain. However, both approaches face a serious knowledge bottleneck, making rapid domain porting difficult. Such systems cannot effectively support user-defined named entities. That is the motivation for using unsupervised or weakly-supervised machine learning that only requires a raw corpus from a given domain for this NE research.

(Cucchiarelli & Velardi 2001) discussed boosting the performance of an existing NE tagger by unsupervised learning based on parsing structures. (Cucerzan & Yarowsky 1999), (Collins & Singer 1999) and (Kim 2002) presented various techniques using co-training schemes for NE extraction seeded by a small list of proper names or handcrafted NE rules. NE tagging has two tasks: (i) NE chunking; (ii) NE classification. Parsing-supported NE bootstrapping systems including ours only focus on NE classification, assuming NE chunks have been constructed by the parser.

The key idea of co-training is the separation of features into several orthogonal views. In case of NE classification, usually one view uses the context evidence and the other relies on the lexicon evidence. Learners corresponding to different views learn from each other iteratively.

One issue of co-training is the error propagation problem in the process of the iterative learning. The rule precision drops iteration-by-iteration. In the early stages, only few instances are available for learning. This makes some powerful statistical

[1] This paper only focuses on classifying proper names. Time and numerical NEs are not yet explored using this method.

models such as HMM difficult to use due to the extremely sparse data.

This paper presents a new bootstrapping approach using successive learning and concept-based seeds. The successive learning is as follows. First, some parsing-based NE rules are learned with high precision but limited recall. Then, these rules are applied to a large raw corpus to automatically generate a tagged corpus. Finally, an HMM-based NE tagger is trained using this corpus. There is no iterative learning between the two learners, hence the process is free of the error propagation problem. The resulting NE system approaches supervised NE performance for some NE types.

To derive the parsing-based learner, instead of seeding the bootstrapping process with NE instances from a proper name list or handcrafted NE rules as (Cucerzan & Yarowsky 1999), (Collins & Singer 1999) and (Kim 2002), the system only requires a few common noun or pronoun seeds that correspond to the concept for the targeted NE, e.g. *he/she/man/woman* for PERSON NE. Such concept-based seeds share grammatical structures with the corresponding NEs, hence a parser is utilized to support bootstrapping. Since pronouns and common nouns occur more often than NE instances, richer contextual evidence is available for effective learning. Using concept-based seeds, the parsing-based NE rules can be learned in one iteration so that the error propagation problem in the iterative learning can be avoided.

This method is also shown to be effective for supporting NE domain porting and is intuitive for configuring an NE system to tag user-defined NE types.

The remaining part of the paper is organized as follows. The overall system design is presented in Section 2. Section 3 describes the parsing-based NE learning. Section 4 presents the automatic construction of annotated NE corpus by parsing-based NE classification. Section 5 presents the string level HMM NE learning. Benchmarks are shown in Section 6. Section 7 is the Conclusion.

2 System Design

Figure 1 shows the overall system architecture. Before the bootstrapping is started, a large raw training corpus is parsed by the English parser from our *InfoXtract* system (Srihari *et al.* 2003). The bootstrapping experiment reported in this paper is based on a corpus containing ~100,000 news articles and a total of ~88,000,000 words. The parsed corpus is saved into a repository, which supports fast retrieval by a keyword-based indexing scheme.

Although the parsing-based NE learner is found to suffer from the recall problem, we can apply the learned rules to a huge parsed corpus. In other words, the availability of an almost unlimited raw corpus compensates for the modest recall. As a result, large quantities of NE instances are automatically acquired. An automatically annotated NE corpus can then be constructed by extracting the tagged instances plus their neighboring words from the repository.

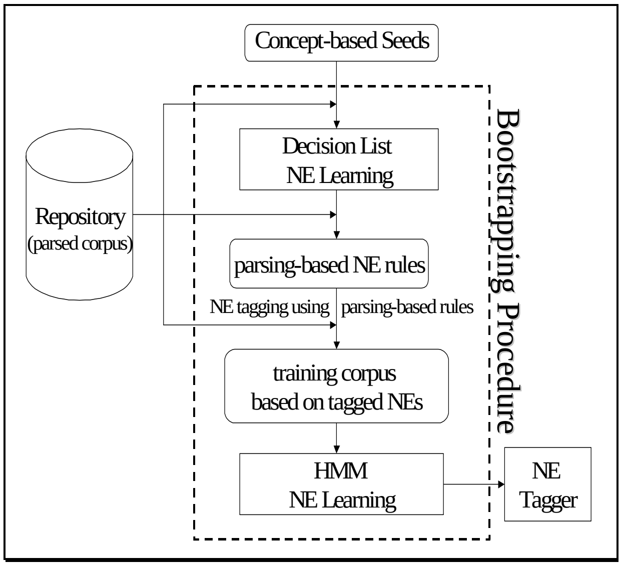

Figure 1. Bootstrapping System Architecture

The bootstrapping is performed as follows:

1. Concept-based seeds are provided by the user.
2. Parsing structures involving concept-based seeds are retrieved from the repository to train a decision list for NE classification.
3. The learned rules are applied to the NE candidates stored in the repository.
4. The proper names tagged in Step 3 and their neighboring words are put together as an NE annotated corpus.
5. An HMM is trained based on the annotated corpus.

3 Parsing-based NE Rule Learning

The training of the first NE learner has three major properties: (i) the use of concept-based seeds, (ii) support from the parser, and (iii) representation as a decision list.

This new bootstrapping approach is based on the observation that there is an underlying concept for any proper name type and this concept can be easily expressed by a set of common nouns or pronouns, similar to how concepts are defined by synsets in WordNet (Beckwith 1991).

Concept-based seeds are conceptually equivalent to the proper name types that they represent. These seeds can be provided by a user intuitively. For example, a user can use *pill, drug, medicine*, etc. as concept-based seeds to guide the system in learning rules to tag MEDICINE names. This process is fairly intuitive, creating a favorable environment for configuring the NE system to the types of names sought by the user.

An important characteristic of concept-based seeds is that they occur much more often than proper name seeds, hence they are effective in guiding the non-iterative NE bootstrapping.

A parser is necessary for concept-based NE bootstrapping. This is due to the fact that concept-based seeds only share pattern similarity with the corresponding NEs at structural level, not at string sequence level. For example, at string sequence level, PERSON names are often preceded by a set of prefixing title words *Mr./Mrs./Miss/Dr.* etc., but the corresponding common noun seeds *man/woman* etc. cannot appear in such patterns. However, at structural level, the concept-based seeds share the same or similar linguistic patterns (e.g. Subject-Verb-Object patterns) with the corresponding types of proper names.

The rationale behind using concept-based seeds in NE bootstrapping is similar to that for parsing-based word clustering (Lin 1998): conceptually similar words occur in structurally similar context. In fact, the anaphoric function of pronouns and common nouns to represent antecedent NEs indicates the substitutability of proper names by the corresponding common nouns or pronouns. For example, *this man* can be substituted for the proper name *John Smith* in almost all structural patterns. Following the same rationale, a bootstrapping approach is applied to the semantic lexicon acquisition task [Thelen & Riloff. 2002].

The InfoXtract parser supports dependency parsing based on the linguistic units constructed by our shallow parser (Srihari *et al.* 2003). Five types of the decoded dependency relationships are used for parsing-based NE rule learning. These are all directional, binary dependency links between linguistic units:

(1) Has_Predicate: from logical subject to verb
e.g. He said she would want him to join. →
he: Has_Predicate(say)
she: Has_Predicate(want)
him: Has_Predicate(join)

(2) Object_Of : from logical object to verb
e.g. This company was founded to provide new telecommunication services. →
company: Object_Of(found)
service: Object_Of(provide)

(3) Has_Amod: from noun to its adjective modifier
e.g. He is a smart, handsome young man. →
man: Has_AMod(smart)
man: Has_AMod(handsome)
man: Has_AMod(young)

(4) Possess: from the possessive noun-modifier to head noun
e.g. His son was elected as mayor of the city. →
his: Possess(son)
city: Possess(mayor)

(5) IsA: equivalence relation from one NP to another NP
e.g. Microsoft spokesman John Smith is a popular man. →
spokesman: IsA(John Smith)
John Smith: IsA(man)

The concept-based seeds used in the experiments are:

1. PER: *he, she, his, her, him, man, woman*
2. LOC: *city, province, town, village*
3. ORG: *company, firm, organization, bank, airline, army, committee, government, school, university*
4. PRO: *car, truck, vehicle, product, plane, aircraft, computer, software, operating system, data-base, book, platform, network*

Note that the last target tag PRO (PRODUCT) is beyond the MUC NE standards: we added this NE type for the purpose of testing the system's capability in supporting user-defined NE types.

From the parsed corpus in the repository, all instances of the concept-based seeds associated with one or more of the five dependency relations are retrieved: 821,267 instances in total in our experiment. Each seed instance was assigned a concept tag corresponding to NE. For example, each instance of *he* is marked as PER. The marked instances plus their associated parsing relationships form an annotated NE corpus, as shown below:

he/PER:	Has_Predicate(say)
she/PER:	Has_Predicate(get)
company/ORG:	Object_Of(compel)
city/LOC:	Possess(mayor)
car/PRO:	Object_Of(manufacture)
HasAmod(high-quality)	
…………	

This training corpus supports the Decision List Learning which learns homogeneous rules (Segal & Etzioni 1994). The accuracy of each rule was evaluated using Laplace smoothing:

$$accuracy = \frac{\text{positive} + 1}{\text{positive} + \text{negative} + \text{NE category No.}}$$

It is noteworthy that the PER tag dominates the corpus due to the fact that the pronouns *he* and *she* occur much more often than the seeded common nouns. So the proportion of NE types in the instances of concept-based seeds is not the same as the proportion of NE types in the proper name instances. For example, considering a running text containing one instance of *John Smith* and one instance of a city name *Rochester*, it is more likely that *John Smith* will be referred to by *he/him* than *Rochester* by (*the*) *city*. Learning based on such a corpus is biased towards PER as the answer.

To correct this bias, we employ the following modification scheme for instance count. Suppose there are a total of N_{PER} PER instances, N_{LOC} LOC instances, N_{ORG} ORG instances, N_{PRO} PRO instances, then in the process of rule accuracy evaluation, the involved instance count for any NE type will be adjusted by the coefficient $\frac{\min(N_{\text{PER}}, N_{\text{LOC}}, N_{\text{ORG}}, N_{\text{PRO}})}{N_{\text{NE}}}$. For example, if the number of the training instances of PER is ten times that of PRO, then when evaluating a rule accuracy, any positive/negative count associated with PER will be discounted by 0.1 to correct the bias.

A total of 1,290 parsing-based NE rules are learned, with accuracy higher than 0.9. The following are sample rules of the learned decision list:

Possess(wife)→ PER
Possess(husband) → PER
Possess(daughter) → PER
Possess(bravery) → PER
Possess(father) → PER
Has_Predicate(divorce) → PER
Has_Predicate(remarry) → PER
Possess(brother) → PER
Possess(son) → PER
Possess(mother) → PER
Object_Of(deport) → PER
Possess(sister) → PER
Possess(colleague) → PER
Possess(career) → PER
Possess(forehead) → PER
Has_Predicate(smile) → PER
Possess(respiratory system) → PER
{Has_Predicate(threaten),
Has_Predicate(kill)} →PER
…………
Possess(concert hall) → LOC
Has_AMod(coastal) → LOC
Has_AMod(northern) → LOC
Has_AMod(eastern) → LOC
Has_AMod(northeastern) → LOC
Possess(undersecretary) → LOC
Possess(mayor) → LOC
Has_AMod(southern) → LOC
Has_AMod(northwestern) → LOC
Has_AMod(populous) → LOC
Has_AMod(rogue) → LOC
Has_AMod(southwestern) → LOC
Possess(medical examiner) → LOC
Has_AMod(edgy) → LOC
…………
Has_AMod(broad-base) → ORG
Has_AMod(advisory) → ORG
Has_AMod(non-profit) → ORG
Possess(ceo) → ORG
Possess(operate loss) → ORG
Has_AMod(multinational) → ORG
Has_AMod(non-governmental) → ORG
Possess(filings) → ORG

Has_AMod(interim) → ORG
Has_AMod(for-profit) → ORG
Has_AMod(not-for-profit) → ORG
Has_AMod(nongovernmental) → ORG
Object_Of(undervalue) → ORG
…………
Has_AMod(handheld) → PRO
Has_AMod(unman) → PRO
Has_AMod(well-sell) → PRO
Has_AMod(value-add) → PRO
Object_Of(refuel) → PRO
Has_AMod(fuel-efficient) → PRO
Object_Of(vend) → PRO
Has_Predicate(accelerate) → PRO
Has_Predicate(collide) → PRO
Object_Of(crash) → PRO
Has_AMod(scalable) → PRO
Possess(patch) → PRO
Object_Of(commercialize)→PRO
Has_AMod(custom-design) → PRO
Possess(rollout) → PRO
Object_Of(redesign) → PRO
…………

Due to the unique equivalence nature of the *IsA* relation, the above bootstrapping procedure can hardly learn *IsA*-based rules. Therefore, we add the following *IsA*-based rules to the top of the decision list: *IsA(seed)→ tag of the seed*, for example:

IsA(man) → PER
IsA(city) → LOC
IsA(company) → ORG
IsA(software) → PRO

4 Automatic Construction of Annotated NE Corpus

In this step, we use the parsing-based first learner to tag a raw corpus in order to train the second NE learner.

One issue with the parsing-based NE rules is modest recall. For incoming documents, approximately 35%-40% of the proper names are associated with at least one of the five parsing relations. Among these proper names associated with parsing relations, only ~5% are recognized by the parsing-based NE rules.

So we adopted the strategy of applying the parsing-based rules to a large corpus (88 million words), and let the quantity compensate for the sparseness of tagged instances. A repository level consolidation scheme is also used to improve the recall.

The NE classification procedure is as follows.

From the repository, all the named entity candidates associated with at least one of the five parsing relationships are retrieved. An NE candidate is defined as any chunk in the parsed corpus that is marked with a proper name Part-Of-Speech (POS) tag (i.e. NNP or NNPS). A total of 1,607,709 NE candidates were retrieved in our experiment. A small sample of the retrieved NE candidates with the associated parsing relationships are shown below:

Deep South : Possess(project)
Ramada : Possess(president)
Argentina : Possess(first lady)
…………

After applying the decision list to the above the NE candidates, 33,104 PER names, 16,426 LOC names, 11,908 ORG names and 6,280 PRO names were extracted.

It is a common practice in the bootstrapping research to make use of heuristics that suggest conditions under which instances should share the same answer. For example, the *one sense per discourse* principle is often used for word sense disambiguation (Gale *et al.* 1992). In this research, we used the heuristic *one tag per domain for multi-word NE* in addition to the *one sense per discourse* principle. These heuristics were found to be very helpful in improving the performance of the bootstrapping algorithm for the purpose of both increasing positive instances (i.e. tag propagation) and decreasing the spurious instances (i.e. tag elimination). The following are two examples to show how the tag propagation and elimination scheme works.

Tyco Toys occurs 67 times in the corpus, and 11 instances are recognized as ORG, only one instance is recognized as PER. Based on the heuristic *one tag per domain for multi-word NE*, the minority tag of PER is removed, and all the 67 instances of *Tyco Toys* are tagged as ORG.

Three instances of *Postal Service* are recognized as ORG, and two instances are recognized as PER. These tags are regarded as noise, hence are removed by the tag elimination scheme.

The tag propagation/elimination scheme is adopted from (Yarowsky 1995). After this step, a total of 386,614 proper names were recognized, including 134,722 PER names, 186,488 LOC names, 46,231 ORG names and 19,173 PRO names. The overall precision was ~90%. The benchmark details will be shown in Section 6.

The extracted proper name instances then led to the construction of a fairly large training corpus sufficient for training the second NE learner. Unlike manually annotated running text corpus, this corpus consists of only sample string sequences containing the automatically tagged NE instances and their left and right neighboring words within the same sentence. The two neighboring words are always regarded as common words while constructing the corpus. This is based on the observation that the proper names usually do not occur continuously without any punctuation in between.

A small sample of the automatically constructed corpus is shown below:

```
in <LOC> Argentina </LOC> .
<LOC> Argentina </LOC> 's
and <PER> Troy Glaus </PER> walk
call <ORG> Prudential Associates </ORG> .
, <PRO> Photoshop </PRO> has
not <PER> David Bonderman </PER> ,
…………
```

This corpus is used for training the second NE learner based on evidence from string sequences, to be described in Section 5 below.

5 String Sequence-based NE Learning

String sequence-based HMM learning is set as our final goal for NE bootstrapping because of the demonstrated high performance of this type of NE taggers.

In this research, a bi-gram HMM is trained based on the sample strings in the annotated corpus constructed in section 4. During the training, each sample string sequence is regarded as an independent sentence. The training process is similar to (Bikel 1997).

The HMM is defined as follows: Given a word sequence $\mathrm{W\,sequence} = \langle w_0 f_0 \rangle \cdots \langle w_n f_n \rangle$ (where f_j denotes a single token feature which will be defined below), the goal for the NE tagging task is to find the optimal NE tag sequence $\mathrm{T\,sequence} = t_0 t_1 t_2 \cdots t_n$, which maximizes the conditional probability $\Pr(\mathrm{T\,sequence} \mid \mathrm{W\,sequence})$ (Bikel 1997). By Bayesian equality, this is equivalent to maximizing the joint probability $\Pr(\mathrm{W\,sequence}, \mathrm{T\,sequence})$. This joint probability can be computed by bi-gram HMM as follows:

$$\begin{aligned}&\Pr(\mathrm{W\,sequence}, \mathrm{T\,sequence})\\ &= \prod_i \Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i\text{-}1}, f_{i\text{-}1} \rangle, t_{i-1})\end{aligned}$$

The back-off model is as follows,

$$\begin{aligned}&\Pr(\langle w_i, f_i \rangle, t_i \mid \langle w_{i\text{-}1}, f_{i\text{-}1} \rangle, t_{i-1})\\ &= \lambda_1 P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i\text{-}1}, f_{i\text{-}1} \rangle, t_{i-1})\\ &+ (1\text{-}\lambda_1)\Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1})\Pr(t_i \mid w_{i-1}, t_{i-1})\end{aligned}$$

$$\begin{aligned}&\Pr(\langle w_i, f_i \rangle \mid t_i, t_{i-1})\\ &= \lambda_2 P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1}) + (1\text{-}\lambda_2)\Pr(\langle w_i, f_i \rangle \mid t_i)\end{aligned}$$

$$\begin{aligned}&\Pr(t_i \mid w_{i\text{-}1}, t_{i-1})\\ &= \lambda_3 P_0(t_i \mid w_{i\text{-}1}, t_{i-1}) + (1\text{-}\lambda_3)\Pr(t_i \mid w_{i\text{-}1})\end{aligned}$$

$$\begin{aligned}&\Pr(\langle w_i, f_i \rangle \mid t_i)\\ &= \lambda_4 P_0(\langle w_i, f_i \rangle \mid t_i) + (1\text{-}\lambda_4)\Pr(w_i \mid t_i) P_0(f_i \mid t_i)\end{aligned}$$

$$\Pr(t_i \mid w_{i\text{-}1}) = \lambda_5 P_0(t_i \mid w_{i\text{-}1}) + (1\text{-}\lambda_5) P_0(t_i)$$

$$\Pr(w_i \mid t_i) = \lambda_6 P_0(w_i \mid t_i) + (1\text{-}\lambda_6)\frac{1}{V}$$

where V denotes the size of the vocabulary, the back-off coefficients λ's are determined using the Witten-Bell smoothing algorithm. The quantities $P_0(\langle w_i, f_i \rangle, t_i \mid \langle w_{i-1}, f_{i-1} \rangle, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i, t_{i-1})$, $P_0(t_i \mid w_{i\text{-}1}, t_{i-1})$, $P_0(\langle w_i, f_i \rangle \mid t_i)$, $P_0(f_i \mid t_i)$, $P_0(t_i \mid w_{i\text{-}1})$, $P_0(t_i)$, and $P_0(w_i \mid t_i)$ are computed by the maximum likelihood estimation.

We use the following single token feature set for HMM training. The definitions of these features are the same as in (Bikel 1997).

twoDigitNum, fourDigitNum, containsDigitAndAlpha, containsDigitAndDash, containsDigitAndSlash, containsDigitAndComma, containsDigitAndPeriod, otherNum, allCaps, capPeriod, initCap, lowerCase, other.

6 Benchmarking and Discussion

Two types of benchmarks were measured: (i) the quality of the automatically constructed NE corpus, and (ii) the performance of the HMM NE tagger. The HMM NE tagger is considered to be the resulting system for application. The benchmarking shows that this system approaches the performance of supervised NE tagger for two of the three proper name NE types in MUC, namely, PER NE and LOC NE.

We used the same blind testing corpus of 300,000 words containing 20,000 PER, LOC and ORG instances that were truthed in-house originally for benchmarking the existing supervised NE tagger (Srihari, Niu & Li 2000). This has the benefit of precisely measuring performance degradation from the supervised learning to unsupervised learning. The performance of our supervised NE tagger using the MUC scorer is shown in Table 1.

Table 1. Performance of Supervised NE Tagger

Type	Precision	Recall	F-Measure
PERSON	92.3%	93.1%	92.7%
LOCATION	89.0%	87.7%	88.3%
ORGANIZATION	85.7%	87.8%	86.7%

To benchmark the quality of the automatically constructed corpus (Table 2), the testing corpus is first processed by our parser and then saved into the repository. The repository level NE classification scheme, as discussed in section 4, is applied. From the recognized NE instances, the instances occurring in the testing corpus are compared with the answer key.

Table 2. Quality of the Constructed Corpus

Type	Precision
PERSON	94.3%
LOCATION	91.7%
ORGANIZATION	88.5%

To benchmark the performance of the HMM tagger, the testing corpus is parsed. The noun chunks with proper name POS tags (NNP and NNPS) are extracted as NE candidates. The preceding word and the succeeding word of the NE candidates are also extracted. Then we apply the HMM to the NE candidates with their neighboring context. The NE classification results are shown in Table 3.

Table 3. Performance of the second HMM NE

Type	Precision	Recall	F-Measure
PERSON	86.6%	88.9%	87.7%
LOCATION	82.9%	81.7%	82.3%
ORGANIZATION	57.1%	48.9%	52.7%

Compared with our existing supervised NE tagger, the degradation using the presented bootstrapping method for PER NE, LOC NE, and ORG NE are 5%, 6%, and 34% respectively.

The performance for PER and LOC are above 80%, approaching the performance of supervised learning. The reason for the low recall of ORG (~50%) is not difficult to understand. For PERSON and LOCATION, a few concept-based seeds seem to be sufficient in covering their sub-types (e.g. the sub-types COUNTRY, CITY, etc for LOCATION). But there are hundreds of sub-types of ORG that cannot be covered by less than a dozen concept-based seeds, which we used. As a result, the recall of ORG is significantly affected. Due to the same fact that ORG contains many more sub-types, the results are also noisier, leading to lower precision than that of the other two NE types. Some threshold can be introduced, e.g. perplexity per word, to remove spurious ORG tags in improving the precision. As for the recall issue, fortunately, in a real-life application, the organization type that a user is interested in usually is in a fairly narrow spectrum. We believe that the performance will be better if only company names or military organization names are targeted.

In addition to the key NE types in MUC, our system is able to recognize another NE type, namely, PRODUCT (PRO) NE. We instructed our truthing team to add this NE type into the testing corpus which contains ~2,000 PRO instances. Table 4 shows the performance of the HMM on the PRO tag.

Table 4. Performance of PRODUCT NE

TYPE	PRECISION	RECALL	F-MEASURE
PRODUCT	67.3%	72.5%	69.8%

Similar to the case of ORG NEs, the number of concept-based seeds is found to be insufficient to cover the variations of PRO subtypes. So the performance is not as good as PER and LOC NEs. Nevertheless, the benchmark shows the system works fairly effectively in extracting the user-specified NEs. It is noteworthy that domain knowledge such as knowing the major sub-types of the user-specified NE type is valuable in assisting the selection of appropriate concept-based seeds for performance enhancement.

The performance of our HMM tagger is comparable with the reported performance in (Collins & Singer 1999). But our benchmarking is more extensive as we used a much larger data set (20,000 NE instances in the testing corpus) than theirs (1,000 NE instances).

7 Conclusion

A novel bootstrapping approach to NE classification is presented. This approach does not require iterative learning which may suffer from error propagation. With minimal human supervision in providing a handful of concept-based seeds, the resulting NE tagger approaches supervised NE performance in NE types for PERSON and LOCATION. The system also demonstrates effective support for user-defined NE classification.

Acknowledgement

This work was partly supported by a grant from the Air Force Research Laboratory's Information Directorate (AFRL/IF), Rome, NY, under contract F30602-01-C-0035. The authors wish to thank Carrie Pine and Sharon Walter of AFRL for supporting and reviewing this work.

References

Bikel, D. M. 1997. Nymble: a high-performance learning name-finder. *Proceedings of ANLP 1997*, 194-201, Morgan Kaufmann Publishers.

Beckwith, R. *et al.* 1991. WordNet: A Lexical Database Organized on Psycholinguistic Principles. *Lexicons: Using On-line Resources to build a Lexicon*, Uri Zernik, editor, Lawrence Erlbaum, Hillsdale, NJ.

Borthwick, A. *et al.* 1998. Description of the MENE named Entity System. *Proceedings of MUC-7*.

Collins, M. and Y. Singer. 1999. Unsupervised Models for Named Entity Classification. *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC.*

Cucchiarelli, A. and P. Velardi. 2001. Unsupervised Named Entity Recognition Using Syntactic and Semantic Contextual Evidence. *Computational Linguistics*, Volume 27, Number 1, 123-131.

Cucerzan, S. and D. Yarowsky. 1999. Language Independent Named Entity Recognition Combining Morphological and Contextual Evidence. *Proceedings of the 1999 Joint SIGDAT Conference on EMNLP and VLC*, 90-99.

Gale, W., K. Church, and D. Yarowsky. 1992. One Sense Per Discourse. *Proceedings of the 4th DARPA Speech and Natural Language Workshop*. 233-237.

Kim, J., I. Kang, and K. Choi. 2002. Unsupervised Named Entity Classification Models and their Ensembles. *COLING 2002*.

Krupka, G. R. and K. Hausman. 1998. IsoQuest Inc: Description of the NetOwl Text Extraction System as used for MUC-7. *Proceedings of MUC-7*.

Lin, D.K. 1998. Automatic Retrieval and Clustering of Similar Words. *COLING-ACL 1998*.

MUC-7, 1998. Proceedings of the Seventh Message Understanding Conference (MUC-7).

Thelen, M. and E. Riloff. 2002. A Bootstrapping Method for Learning Semantic Lexicons using Extraction Pattern Contexts. *Proceedings of EMNLP 2002*.

Segal, R. and O. Etzioni. 1994. Learning decision lists using homogeneous rules. *Proceedings of the 12th National Conference on Artificial Intelligence*.

Srihari, R., W. Li, C. Niu and T. Cornell. 2003. InfoXtract: An Information Discovery Engine Supported by New Levels of Information Extraction. *Proceeding of HLT-NAACL 2003 Workshop on Software Engineering and Architecture of Language Technology Systems*, Edmonton, Canada.

Srihari, R., C. Niu, & W. Li. 2000. A Hybrid Approach for Named Entity and Sub-Type Tagging. *Proceedings of ANLP 2000*, Seattle.

Yarowsky, David. 1995. Unsupervised Word Sense Disambiguation Rivaling Supervised Method. *ACL 1995*.