Unsupervised Named Entity Classification Models and their Ensembles

Jae-Ho Kim*, In-Ho Kang, Key-Sun Choi*

Korea Advanced Institute of Science and Technology (KAIST) /
Korea Terminology Research Center for Language and Knowledge Engineering* (KORTERM)
373-1, Guseong-dong, Yuseong-gu
Daejeon, KOREA, 305-701
{jjaeh@world, ihkang@csone, kschoi@world}.kaist.ac.kr

Abstract

This paper proposes an unsupervised learning model for classifying named entities. This model uses a training set, built automatically by means of a small-scale named entity dictionary and an unlabeled corpus. This enables us to classify named entities without the cost for building a large hand-tagged training corpus or a lot of rules.

Our model uses the ensemble of three different learning methods and repeats the learning with new training examples generated through the ensemble learning. The ensemble of various learning methods brings a better result than each individual learning method. The experimental result shows 73.16% in precision and 72.98% in recall for Korean news articles.

1 Introduction

Named entity extraction is an important step for various applications in natural language processing. Named entity extraction involves identifying named entities in the text and classifying their types such as person, organization, location, time expressions, numeric expressions, and so on (Sekine and Eriguchi, 2000).

One might think the named entities can be classified easily using dictionaries because most of named entities are proper nouns, but this is wrong opinion. As time passes, new proper nouns are created continuously. Therefore it is impossible to add all those proper nouns to a dictionary. Even though named entities are registered in the dictionary it is not easy to decide their senses. They have a semantic (sense) ambiguity that a proper noun has different senses according to the context (Nina Wacholder, *et al*., 1997). For example, 'United States' refers either to a geographical area or to the political body which governs this area. The semantic ambiguity is occured frequently in Korean (Seon, *et al*. 2001). Let us illustrate this.

Example 1 : Location
Let's meet at KAIST.

KAIST	***e-seo***	*man-na-ja* .
(PN:KAIST)	(PP:at)	(V:meet)

Example 2 : Organization
KAIST announced the list of successful candidates.

KAIST	***e-seo***	*hab-gyeok-ja*
(PN:KAIST)	(PP)	(N:successful candidates)

myeong-dan	*eul*	*bal-pyo-haet-da* .
(N:list)	(PP)	(V:announced)

PN : proper noun, N : noun, PP : postposition, V : verb

In the above examples, 'KAIST' has different categories although same postposition, '*e-seo*', followed. The classification of named entities in Korean is a little more difficult than in English.

There are two main approaches to classify named entities. The first approach employs hand-crafted rules. It costs too much to maintain rules because rules and dictionaries have to be changed according to the application. The second belongs to a supervised learning approach, which employs a statistical method. As it is more robust and requires less human intervention, several statistical methods based on a hidden Markov model (Bikel *et al*., 1997), a

Maximum Entropy model (Borthwich *et al*., 1998) and a Decision Tree model (Béchet *et al*. 2000) have been studied. The supervised learning approach requires a hand-tagged training corpus, but it can not achieve a good performance without a large amount of data because of data sparseness problem. For example, Borthwich (1999) showed the performance of 83.45% in Precision and 77.42% in F-measure for identifying and classifying the 8 IREX (IREX committee, 1999) categories, with 294,000 tokens IREX training corpus. It takes a lot of time and labor to build a large corpus like this.

This paper proposes an unsupervised learning model that uses a small-scale named entity dictionary and an unlabeled corpus for classifiying named entities. Collins and Singer (1999) opened the possibility of using an unlabeled corpus to classify named entities. They showed that the use of unlabeled data can reduce the requirements for supervision to just 7 simple seed rules. They used natural redundancy in the data : for many named-entity instances, both the spelling of the name and the context in which it appears are sufficient to determine its type.

Our model considers syntactic relations in a sentence to resolve the semantic ambiguity and uses the ensemble of three different learning methods to improve the performance. They are Maximum Entropy Model, Memory-based Learning and Sparse Network of Winnows (Roth, 1998).

This model classifies proper nouns appeared in the documents into person, organization and location on the assumption that the boundaries of proper nouns were already recognized.

2 The System for NE Classification

This section describes a system that classifies named entities by using a machine learning algorithm. The system consists of four modules as shown in Figure 1.

First, we builds a training set, named entity tagged corpus, automatically. This set will be used to predict the categories of named entities within target documents received as the input of the system.

The second module extracts syntactic relations from the training set and target documents. They are encoded to the format of training and test examples for machine learning.

In the third module, each learning for classification is progressed independently by three learning methods. Three results generated by each learner are combined into one result.

Finally, the system decides the category by using a rule for the test examples that did not be labeled yet. And then the system outputs a named entity tagged corpus.

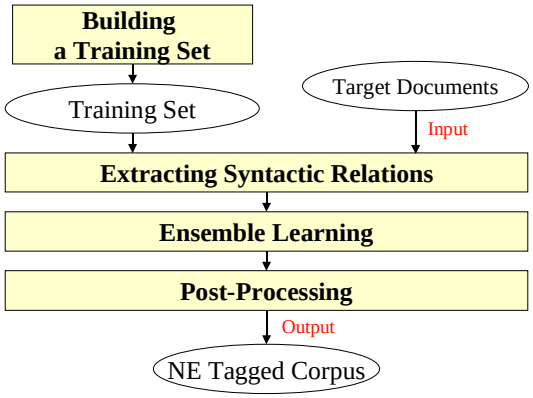

Figure 1. System Architecture

2.1 Building a Training Set

The system requires a training set which has categories in order to get knowledge for the classification. We build a training set automatically using a named entity dictionary and a POS tagged corpus, and then use it instead of a hand-tagged set in machine learning.

We randomly extract 1500 entries per each category (person, location, and organization) from a Proper Noun dictionary made by KORTERM and then reconstruct the named entity dictionary. The Proper Noun dictionary has about 51,000 proper nouns classified into 41 categories (person, animal, plant and etc.). We do not extract homonyms to reduce the ambiguity. In order to show that it is possible to classify named entities with a small-scale dictionary, we limit the number of entries to be 1500.

We label the target word, proper noun or capital alphabet, appeared in the POS tagged corpus [1] by means of the NE dictionary mentioned above. The corpus is composed of

[1] We used a KAIST POS tagged corpus

one million *eojeols*[2]. It is not easy to classify named entity correctly only with a dictionary, since named entity has the semantic ambiguity. So we have to consider the context around the target word.

In order to consider the context, we use co-occurrence information between the category (*c*) of a target word (*tw*) and a head word (*hw*) appeared on the left of the target word or the right of the target word. We modify categories labeled by the NE dictionary by following process.

1. We extract pairs [*c*, *hw*] from the corpus labeled by means of the dictionary.
2. If *hw* is occurred with several different categories, we suppose *tw* occurred with *hw* may have an ambiguity and then we remove the category label of *tw*.
3. We make rules for predicting the category of *tw* from pairs [*c*, *hw*] and apply them to the corpus. The rule is that *tw* occurred with *hw* has a *c*.
4. We extract sentences including the labeled target word in the corpus.

In the step 3, 9 rules are made. We label the *c* for unlabeled target word occurred with *hw* if the pair [*c*, *hw*] is found more than a threshold. We set the threshold to be 10. Sentences including the 4,504 labeled target word are made as a tringing set in this process (Table 1).

Table 1. The number of the target words in a training set

State	# of target words
Candidates in the corpus	*37,831*
Labeled by the dictionary	3,899
Removed by the ambiguity	778
Added by 9 rules	1,383
Total	4,504

2.2 Extracting Syntactic Relations

In order to predict the category, most of machine learning systems usually consider two words on the left and two ones on the right of a target word as a context (Uchimoto and *et al.* 2000, Petasis and *et al.* 2000). However this method have some problems.

If some words that are not helpful to predict the category are near the target word, they can cause an incorrect prediction. In the following example, '*Kim*' can be predicted as an organization instead of a person because of a left word '*Jeong-bu*' (the government).

Example
The goverment supports KIA on the premise that the chairman Kim submits a resignation.

Jeong-bu *neun* ***Kim*** *hoi-jang* *i*
(N:the goverment) (PP) (PN) (N:the chairman) (PP)

sa-pyo *reul* *je-chul-han-da* *neun*
(N:a resignation) (PP) (V :submit) (PP)

jeon-je *ro* ***KIA*** *reul* *ji-won-han-da.*
(N :the premise)(PP) (PN) (PP) (V :support)

PN : proper noun, N : noun, PP : postposition, V : verb

The system cannot consider important words that are out of the limit of the context. In the former example, the word '*je-chul-han-da*' (submit) is an important feature for predicting the category of '*Kim*'. If a Korean functional word is counted as one window, we cannot get this information within right 4 windows. Even if we do not count the functional words, sometimes it is neccessary to consider larger windows than 2 windows like above example.

We notice that words that modify the target word or are modified by the target word are more helpful to the prediction than any other words in the sentence. So we extract the syntactic relations like Figure 2 as the context.

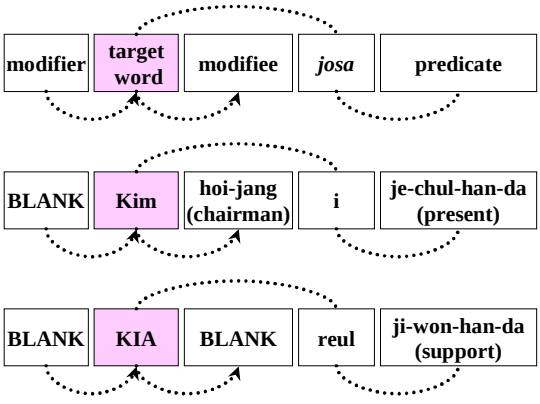

Figure 2. Syntactic relations for the target word

The modifier is a word modifying the target word and the modifiee is one modified by the

[2] Korean linguistic units that is separated by blank or punctuation

target word. *Josa*[3] is a postposition that follows the target word and te predicate is a verb that predicates the target word. The 'BLANK' label represents that there is no word which corresponds to the slot of the templet. These syntactic relations are extracted by a simple heuristic parser. We will show that these syntactic relations bring to a better result through an experiment in the section 3.

These syntactic relations seem to be language specific. *Josa* represents a case for the target word. If case information is extracted in a sentence, these syntactic relations like Figure 2 are also made in other languages.

As machine learner requires training and test examples represented in a feature-vector format, syntactic relations are encoded as Figure 3.

Feature-vector format

Modifier	lexical morpheme (*w*)
	POS tag (*t*)
Target word	lexical morpheme (*w*)
	POS tag (*t*)
Modifiee	lexical morpheme (*w*)
	POS tag (*t*)
Josa	lexical morpheme (*w*)
Predicate	lexical morpheme (*w*)
Category	Label tag

Training example : [*w, t, w, t, w, t, w, w, person*]
Test example : [*w, t, w, t, w, t, w, w, Blank*]

Figure 3: The format of an example for learning

2.3 Ensemble Learning

The ensemble of several classifiers can be improve the performance. Errors made by the minority can be removed through the ensemble of classifiers (Thomas G. Dietterich, 1997). In the base noun phrase identification, Tjong Kim Sang, *et al*. (2000) showed that the result combined by seven different machine learning algorithms outperformed the best individual result.

In our module, machine learners train with the training examples and then classify the named entities in the test examples. This process is shown in Figure 4.

[3] Josa, attached to a nominal, is a postpositional particle in Korean.

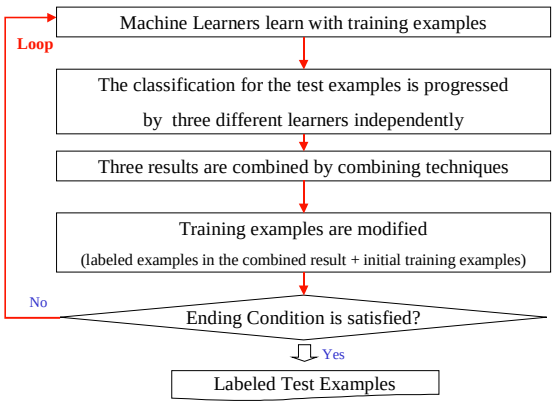

Figure 4. The process of the Ensemble Learning

This ensemble learning has two characteristics. One is that the classification is progressed by three different learners independently and those results are combined into one result. The other is that the learning is repeated with new training examples generated through the learning. It enables the system to receive an incremental feedback.

Through the this learning method, we can get larger and more precise training examples for predicting the categories. It is important in an unsupervisd learning model because there is no labeled data for learning.

2.3.1 Machine Learning algorithms

We use three learning methods : Memory-based Learning, Sparse Network of Winnows, Maximum Entropy Model. We describe these methods briefly in this section.

Memory-based Learning stores the training examples and classifies new examples by choosing the most frequent classification among training examples which are closest to a new example. Examples are represented as sets of feature-value pairs. Each feature receives a weight which is based on the amount of information which it provides for computing the classification of the examples in the training data. We use the TiMBL (Daelemans, *et al*., 1999), a Memory-Based Learning software package.

Sparse Network of Winnows learning architecture is a sparse network of linear units. Nodes in the input layer of the network represent simple relations over the input example and things being used as the input features. Each linear unit is called a target node and represents

classifications which are interested in the input examples. Given training examples, each input example is mapped into a set of features which are active (present) in it; this representation is presented to the input layer of SNoW and propagated to the target nodes. We use SnoW (Carlson, *et al*., 1999), Sparse Network of Winnows software package.

Maximum Entropy Model (MEM) is especially suited for integrating evidences from various information sources. MEM allows the computation of *p(f|h)* for any *f* in the space of possible futures, F, and for every *h* in the space of possible histories, H. Futures are defined as the possible classification and a history is all of the conditioning data which enable us to make a decision in the space of futures. The computation of *p(f|h)* is dependent on a set of features which are binary functions of the histroy and future. A feature is represented as following.

$$g(h, f) = \begin{cases} 1 & \text{if h meets some condition} \\ & \text{and f is one of the future} \\ 0 & \text{otherwise} \end{cases}$$

Given a set of features and some training examples, a weighing parameter α_i for every feature g_i is computed. This allows us to compute the conditional probability as follows :

$$P(f \mid h) = \frac{\prod_i \alpha_i^{g_i(h,f)}}{Z_\alpha(h)}$$

$$Z_\alpha(h) = \sum_f \prod_i \alpha_i^{g_i(h,f)}$$

We use MEMT, Maximum Entropy Modeling Toolkit (Ristad, 1998), to compute the parameter for the features.

2.3.2 Combining Techniques

We use three different voting mechanisms to combine results generated by three learners.

The first method is a majority voting. Each classification receives the same weight and the most frequent classification is chosen. The ending condition is satisfied when there is no difference between a result combined in this loop and one combined in the former loop.

The second method is a probability voting. MEMT and SNoW propose the probabilities for all category, but Timbl proposes only one appropriate category for one test example. We set the probability for the category Timbl proposes to be 0.6 and for the others to be 0.2. For each category, we multiply probabilities proposed by 3 learners and then choose N examples that have the largest probability. In the next learning we set N = N + 100. When N is larger than a threshold, the ending condition is satisfied and the learning is over. We set it to be 3/4 of the number of test examples.

The last method is a mixed voting. We use two voting methods mentioned above one after another. First, we use probability voting. After the learning is over we use majority voting. The threshold of the probability voting is 1/2 of the number of test examples here.

2.4 Post-Processing

After the learning, the system modifies test examples by using a rule, one sense per discourse. One sense per discourse means that the sense of a target word is highly consistent within any given document. David Yarowsky (1995) showed it was accurate in the word sense disambiguation. We label the examples that are not labeled yet as the category of the labeled word in the discourse as following example and we output named entity tagged corpus.

Example

after the ensemble learning

... ... ***KIA*****<type=organization>** *reul ji-won-han-da.* ***KIA*** *neon* **... ...**

after post-processing

... ... ***KIA*****<type=organization>** *reul ji-won-han-da.* ***KIA*****<type=organization>** *neon* **... ...**

3 Experimental Results

We used Korean news articles that consist of 24,647 *eojeols* and contain 2,580 named entities as a test set. The number of named entities which belong to each category is shown in Table 2. When even a human could not classify named entities, 'Unknown' is labeled and it is ignored for the evaluation. 'Other' is used for the word outside the three categories.

Table 3 shows the result of the classification. The first row shows the result of the classification using only a NE dictionary. The recall (14.84%) is very low because the system

uses a small-scale dictionary. The precision (91.56%) is not 100% because of the semantic ambiguity. It means that it is necessary to refine classifications created by a dictionary.

We build a training set with a NE dictionary and a POS tagged corpus and refine it with co-occurrence information. The second row shows the result of the classification using this training set without learning. We can observe that the quality of the training set is improved thanks to our refining method.

A Mixed Voting shows the best results. It improves the performance by taking good characteristics of a majority voting and probability voting.

Table 2. The number of named entities which belong to each category in the test set

Category	# of NEs	Category	# of NEs
Person	459	Other	307
Organization	814	Unknown	242
Location	758	Total	2,580

Table 3. The result of the classification

Method	Precision	Recall	F-measure
Dictionary based	91.56%	14.84%	25.54%
Training set b1ased	94.32%	20.64%	33.87%
Majority Voting	69.70%	65.74%	67.68%
Probability Voting	75.90%	63.45%	69.12%
Mixed Voting	**73.16%**	**72.98%**	**73.07%**

We extract the syntatic relations and make 5 windows (modifier, target word, modifiee, *josa*, predicate) as a context. We conduct a comparative experiment using the Uchimoto's method, 5 windows (two words before/after the target word) and then we show that our method brings to a better result (Table 4).

Table 4. Comparison with two kinds of window size

Windows	Precision	Recall	F-measure
Uchimoto's	66.86%	69.94%	68.37%
Ours	**73.16%**	**72.98%**	**73.07%**

We try to perform the co-training similar to one of Colins and Singer in the same experimental environment. We extract contextual rules from our 5 windows because we does not have a full parser. The learning is started from 417 spelling seed rules made by the NE dictionary. We use two independent context and spelling rules in turn. Table 5 shows that our method improve the recall much more on the same conditions.

Table 5. Comparison with two kinds of unsupervised learning method

Method	Precision	Recall	F-measure
Co-trianing	84.62%	37.63%	52.09%
Ours	**73.16%**	**72.98%**	**73.07%**

Through the ensemble of various learning methods, we get larger and more precise training examples for the classification. Table 6 shows that the ensemble learning brings a better result than each individual learning method.

Table 6. The comparison of an ensemble learning and each individual learning

Learner	Precision	Recall	F-measure
MEMT	65.19%	61.54%	63.31%
SNoW	66.93%	70.53%	68.68%
Timbl	64.14%	67.59%	65.82%
Ensemble	**73.16%**	**72.98%**	**73.07%**

Three learners can use different kinds of features instead of same features. We conduct a comparative experiment as following. As features, SNoW uses a modifier and a target word, Timbl uses a modifiee and a target word, and MEMT uses a *josa*, a predicate and a target word. Table 7 shows that the learning using different kinds of features has the low performance because of the lack of information.

Table 7. The comparison with the learnings using different features

Features	Precision	Recall	F-measure
Seperated	61.69%	49.85%	55.14%
Same	**73.16%**	**72.98%**	**73.07%**

The system repeats the learning with new training examples generated through the ensemble learning. We can see that this loop brings to the better result as shown in Table 8.

After the learning, we apply the rule, a sense per discourse. 'Post' in Table 8 indicates the performance after this post-processing. It The

post-processing improves the performance a little.

Table 8. The improvement of the performance through the repeated learning

Method	Loop	Precision	Recall	F-measure
Probability Voting	1st	94.35%	20.76%	34.03%
	19th	76.72%	59.97%	67.32%
	Post	**75.90%**	**63.45%**	**69.12%**

We extracted the syntactic relations by using a simple heuristic parser. Because this parser does not deal with complex sentences, the failure of parsing causes the lack of information or wrong learning. Most of errors are actually occurred by it, therefore we need to improve the performance of the parser.

4 Conclusion

We proposed an unsupervised learning model for classifying the named entities. This model used a training set, built automatically by a small-scale NE dictionary and an unlabeled corpus, instead of a hand-tagged training set for learning. The experimental result showed 73.16% in precision and 72.98% in recall for Korean news articles. This means that it is possible to classify named entities without the cost for building a large hand-tagged training corpus or a lot of rules.

The learning for classification was progressed by the ensemble of three different learning methods. Then the ensemble of various learning methods brings a better result than each individual learning method.

References

Béchet, Frédéric, Alexis Nasr and Franck Genet, 2000. "Tagging Unknown Proper Names Using Decision Trees", In proceedings of the 38th Annual Meeting of the Association for Computational Linguistics.

Bikel, Daniel M., Scott Miller, Richard Schwartz and Ralph Weischedel, 1997. "Nymble: a High-Performance Learning Name-finder", In Proceedings of the Fifth Conference on Applied Natural Language Processing.

Borthwick, Andrew, John Sterling, Eugene Agichtein and Ralph Grishman, 1998. "NYU: Description of the MENE Named Entity System as Used in MUC-7", In Proceedings of the Seventh Message Understanding Conference (MUC-7).

Borthwick, 1999. “A Japanese Named Entity Recognizer Constructed by a Non-Speaker of Japanese”, IREX. Proceedings of the IREX workshop.

Carlson, Andrew J., Chad M. Cumby, Jeff L. Rosen and Dan Roth, 1999. "SNoW User Guide", University of Illinois. http://l2r.cs.uiuc.edu/~cogcomp/

Collins, Michael and Yoram Singer. 1999. "Unsupervised models for named entity classification", In proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora.

Daelemans, Walter, Jakub Zavrel, Ko van der Sloot and Antal van den Bosch, 1999. "TiMBL: Tilburg Memory Based Learner, version 4.0, Reference Guide", ILK Technical Report 01-04. http://ilk.kub.nl/

Dietterich, T. G., 1997. “Machine-Learning Research: Four Current Dirctions”, AI Magazine 18(4): 97

IREX Committee (ed.), 1999. Proc. the IREX Workshop. http://cs.nyu.edu/cs/projects/proteus/irex

Petasis, Georgios, Alessandro Cucchiarelli, Paola Velardi, Georgios Paliouras, Vangelis Karkaletsis and Constantine D. Spyropoulos, 2000. "Automatic adaptation of Proper Noun Dictionaries through cooperation of machine learning and probabilistic methods", Proceedings of the 23rd ACM SIGIR Conference on R&D in IR (SIGIR).

Ristad, Eric Sven, 1998. "Maximum Entropy Modeling Toolkit".

Roth, Dan, 1998. “Learning to resolve natural language ambiguities: A unified approach”, In Proc. National Conference on Artificial Intelligence.

Sekine, Satoshi and Yoshio Eriguchi. 2000. "Japanese Named Entity Extraction Evaluation", In the proceedings of the 18th COLING.

Seon, Choong-Nyoung, Youngjoong Ko, Jeong-Seok Kim and Jungyun Seo, 2001. “Named Entity Recognition using Machine Learning Methods and Pattern-Selection Rules”, Proceedings of the Sixth Natural Language Processing Pacific Rim Symposium.

Tjong Kim Sang, Erik F., Walter Daelemans, Hervé Déjean, Rob Koeling, Yuval Krymolowski, Vasin Punyakanok, Dan Roth, 2000. “Applying System Combination to Base Noun Phrase Identification”, In the proceedings of the 18th COLING.

Uchimoto, Kiyotaka, Qing Ma, Masaki Murata, Hiromi Ozaku and Hitoshi Isahara, 2000. “Named Entity Extraction Based on A Maximum Entropy Model and Transformation Rules", In proceedings of the 38th Annual Meeting of the Association for Computational Linguistics.

Wacholder, Nina, Yael Ravin and Misook Choi (1997) "Disambiguation of Proper Names in Text", Proceedings of the 5th Applied Natural Language Processing Conference.

Yarowsky, David, 1995. "Unsupervised Word Sense Disambiguation Rivaling Supervised Methods", In Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics.