Extended Named Entity Recognition API and Its Applications in Language Education

Nguyen Tuan Duc[1], Khai Mai[1], Thai-Hoang Pham[1], Nguyen Minh Trung[1], Truc-Vien T. Nguyen[1], Takashi Eguchi[1], Ryohei Sasano[2], Satoshi Sekine[3]
[1]Alt Inc [2]Nagoya University [3]New York University
nguyen.tuan.duc@alt.ai, sekine@cs.nyu.edu

Abstract

We present an Extended Named Entity Recognition API to recognize various types of entities and classify the entities into 200 different categories. Each entity is classified into a hierarchy of entity categories, in which the categories near the root are more general than the categories near the leaves of the hierarchy. This category information can be used in various applications such as language educational applications, online news services and recommendation engines. We show an application of the API in a Japanese online news service for Japanese language learners.

1 Introduction

Named entity recognition (NER) is one of the most fundamental tasks in Information Retrieval, Information Extraction and Question Answering (Bellot et al., 2002; Nadeau and Sekine, 2007). A high quality named entity recognition API (Application Programming Interface) is therefore important for higher level tasks such as entity retrieval, recommendation and automatic dialogue generation. To extend the ability of named entity recognition, Sekine et al. (Sekine et al., 2002; Sekine and Nobata, 2004) have proposed an Extended Named Entity (ENE) hierarchy, which refines the definition of named entity. The ENE hierarchy is a three-level hierarchy, which contains more than ten coarse-grained categories at the top level and 200 fine-grained categories at the leaf level.

The top level of the hierarchy includes traditional named entity categories, such as Person, Location or Organization. The middle level and leaf level refine the top level categories to more fine-grained categories. Figure 1 shows a partial hierarchy for the top level category *Organization*. In Extended Named Entity recognition (ENER) problem, given an input sentence, such as "Donald Trump was officially nominated by the Republican Party", the system must recognize and classify the ENEs in the sentence, such as "Donald Trump" as *Person* and "Republican Party" as *Political Party*.

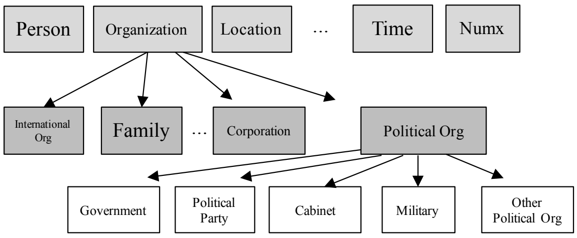


Figure 1: Extended Named Entity hierarchy

In this paper, we present the architecture design and implementation of an ENER API for Japanese. We named this API as "AL+ ENER API". The proposed architecture works well with a large number of training data samples and responses fast enough to use in practical applications. To illustrate the effectiveness of the AL+ ENER API, we describe an application of the API for automatic extraction of glossaries in a Japanese online news service for Japanese language learners. Feedbacks from the users show that the presented ENER API gives high precision on the glossary creation task.

The rest of this paper is organized as follows. Section 2 describes the design and implementation of the ENER API. Experiment results are presented in Section 3 to evaluate the performance of the API. Section 4 describes an application of the ENER API into an online news service for Japanese learners, the method to get user feedbacks from this service to improve the ENER system, and the statistics obtained from the user feed-

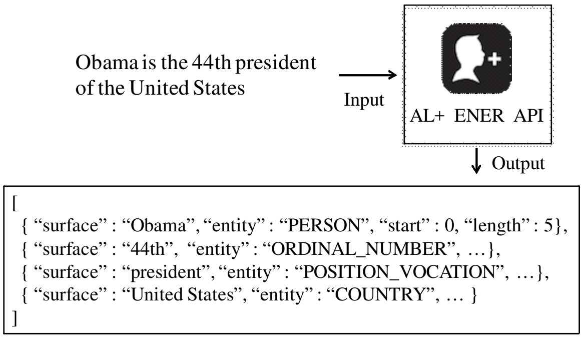
Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics-System Demonstrations, pages 37–42
Vancouver, Canada, July 30 - August 4, 2017. ©2017 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17-4007

backs. Section 5 reviews related systems and compares with the presented system. Finally, Section 6 concludes the paper.

2 Extended Named Entity Recognition API

2.1 Overview of the AL+ ENER API

The AL+ ENER API is an API for Extended Named Entity recognition, which takes an input sentence and outputs a JSON containing a list of ENEs in the sentence, as shown in Figure 2.



```
[
  { "surface" : "Obama", "entity" : "PERSON", "start" : 0, "length" : 5},
  { "surface" : "44th",  "entity" : "ORDINAL_NUMBER", ...},
  { "surface" : "president", "entity" : "POSITION_VOCATION", ...},
  { "surface" : "United States", "entity" : "COUNTRY", ... }
]
```

Figure 2: AL+ ENE Recognition API

Different from traditional NER APIs, this ENER API is capable of tagging 200 categories[1], including some entities that are actually not named entities (therefore, they are called "extended" named entities, as described in (Sekine and Nobata, 2004)). In Figure 2, "president" is not a traditional named entity, but it is tagged as POSITION_VOCATION, which is a category in the ENE hierarchy. For each entity, we output its surface (e.g., "president"), its ENE tag ("POSITION_VOCATION"), its index in the input sentence (the "start" field in the JSON) and its length. A developer who uses the ENER API can utilize the start and length information to calculate the exact position of the entity in the input sentence. The ENE tag can then be used in various subsequent tasks such as Relation Extraction (RE), Question Answering (QA) or automatic dialogue generation. The AL+ ENER API is freely accessible online.[2] Currently, the API supports Japanese only, but we are also developing an API for English ENER. Figure 3 shows an example input sentence and output ENE tags.

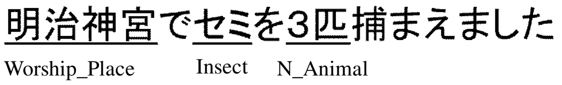


Figure 3: An example input sentence and output ENE tags. Translated sentence with tags: "I caught 3/N_Animal cicadas/Insect at Meiji Shrine/Worship_Place".

2.2 Extended Named Entity recognition algorithms

Existing NER systems often use Conditinal Random Fields (CRFs) (McCallum and Li, 2003; Finkel et al., 2005), HMM (Zhou and Su, 2002) or SVM (Yamada et al., 2002; Takeuchi and Collier, 2002; Sasano and Kurohashi, 2008) to assign tags to the tokens in an input sentence. However, these methods are supposed to work with only small number of categories (e.g., 10 categories). In the ENER problem, the number of categories is 200, which is very large, compared with the number in traditional NER. Consequently, traditional approaches might not achieve good performance and even be infeasible. Actually, we have tried to use CRF for 200 classes, but the training process took too long time and did not finish.

In this system, we use a combination approach to recognize ENEs. We first implement four base algorithms, namely, CRF-SVM hierarchical ENER, RNN-based ENER, Wikification-based ENER and Rule-based ENER. We then combine these algorithms by a selection method, as shown in Figure 4.

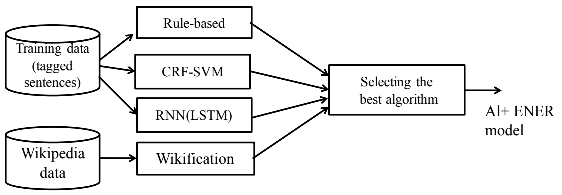


Figure 4: Overview of the proposed ENER algorithm

In the Rule-based method, we extend the rule-based method in (Sekine and Nobata, 2004) (by adding new rules for the new categories that are not recognized in their work) and we also use a dictionary containing 1.6 million Wikipedia entities. In the 1.6 million entities in the dictionary, only 70 thousands entities are assigned ENE tags by human, the rest are assigned by an existing Wikipedia ENE labeling algorithm (Suzuki et al.,

[1] The list of categories is here: http://nlp.cs.nyu.edu/ene/
[2] http://enerdev.alt.ai:8030/#!/Chatbot/

2016), which gives a score for each (entity, ENE category) pair. For the entities that are assigned automatically, we only take the entities with high scores to ensure that the algorithm assigns correct labels. If the rules fail to extract some entities, we extract all noun-phrases and lookup in the dictionary to check if they can be ENEs or not.

We use a training dataset which contains ENE-tagged sentences to train a CRF model to tag input sentences with the top-level ENE categories (in the training dataset, we get the correct labels for these ENEs from the parent or grandparent category in the ENE hierarchy). As illustrated in Figure 1, at the top level, we only have 11 ENE categories that we need to recognize by CRF-SVM (other categories such as Date, Time, Number can be recognized by rules), thus using a CRF model here would achieve comparable performance with existing NER systems. After tagging the sentences with the top-level ENE categories, we can convert the ENER problem into a simple classification problem (not a sequence labeling problem anymore), thus we can use SVM to classify the extracted ENEs at the top level into leaf-level categories. Therefore, we have a CRF model to tag the input sentences with top-level categories, and several SVM models (each for a top-level category) to classify the ENEs into the leaf-level ENE categories. The features that we use in CRF and SVM are bag-of-words, POS-tag, the number of digits in the word, the Brown cluster of the current word, the appearance of the word as a substring of a word in the Wikipedia ENE dictionary, the orthography features (the word is written in Kanji, Hiragana, Katakana or Romaji), whether the word is capitalized, and the last 2-3 characters. Because the number of leaf-level categories in each top-level category is also not too large (e.g., less than 15), SVM can achieve a reasonable performance at this step.

We also train an LSTM (Long-Short Term Memory network), a kind of RNN (Recurrent Neural Network) to recognize ENEs. We use LSTM because it is appropriate for sequence labeling problems. The inputs of the LSTM are the word embedding of the current word and the POS-tag of the current word. The POS-tags are automatically generated using JUMAN[3], a Japanese morphological analyzer. The word embedding is obtained by training a word2vec model with Japanese Wikipedia text. We hope that LSTM can memorize the patterns in the training data and interpolate to the CRF-SVM method in many cases.

To cope with free-text ENEs, we use Wikification approach. Free-text ENEs refer to the entities that can be of any text, such as a movie name or a song name (e.g., "What is your name" is a famous movie name in Japanese). If these names are famous, they often become the titles of some Wikipedia articles. Consequently, using Wikification-based approach could work well with these types of entities.

We also create an algorithm selection model by evaluating the F-scores of the four base algorithms (Rule, CRF-SVM, RNN and Wikification) with a development dataset (which is different from the test set). In the final phase, after having all labels from the four base algorithms for each entity, we select the label of the algorithm with the highest F-score in the development set. Note that we use the best selection scheme at entity level, not at sentence level. This is because each base algorithm tends to achieve high performance on some specific categories, so if we select the best algorithm for each entity, we will achieve higher performance for the entire sentence.

3 Evaluation

3.1 Data set

We hired seven annotators to create an ENE tagged dataset. Specifically, for each ENE category, the annotators created 100 Japanese sentences, each sentence includes at least one entity in the corresponding category. The annotators then manually tagged the sentences with ENE tags. After filtering out erroneous sentences (sentences with invalid tag format), we obtain totally 19,363 well-formed sentences. We divided the dataset into three subsets: the training set (70% of the total number of sentences), development set (15%) and test set (15%). Table 1 shows some statistics of the dataset.

Dataset	No. sentences	No. tokens	No. entities
Train	13,625	266,454	37,062
Dev	2,869	58,529	7,673
Test	2,869	55,999	7,711

Table 1: Statistics of the datasets

[3] http://nlp.ist.i.kyoto-u.ac.jp/EN/?JUMAN

3.2 Performance of the ENER API

We use the test set to evaluate the precision, recall and F-score of the ENER API. Table 2 shows the Precision, Recall and F-score of the ENER API on some specific categories as well as the average evaluation results of the entire 200 categories (in the last row). We achieved very high performance on the categories with small number of known entities (such as Cabinet) or the categories that the rules can capture almost all entities (such as Intensity, Volume, URL, and Email). For categories with free text names (e.g, printing names) or very short name (e.g., AK-47, a type of weapon) the system can not predict the ENE very well because these names might appear in various contexts. We might prioritize Wikification method in these cases to improve the performance. On average, we achieve an F1-score of 71.95%, which is a reasonable result for 200 categories.

Category	Precision (%)	Recall (%)	F-score (%)
Cabinet	100.00	100.00	100.00
Intensity	100.00	100.00	100.00
URL	100.00	100.00	100.00
Phone_Number	100.00	95.25	97.56
Email	100.00	93.33	96.55
Volume	100.00	93.10	96.43
...	...	...	...
Aircraft	80.95	65.38	72.34
Company_Group	68.42	76.47	72.22
Continental_Region	74.29	69.33	71.72
...	...	...	...
Printing_Other	50.00	11.76	19.05
Name_Other	23.08	15.00	18.18
Weapon	9.09	4.17	5.71
Average	**73.47**	**70.50**	**71.95**

Table 2: Precision, Recall, F-score of the ENER API on the test dataset

3.3 Response time of the API

As ENER is often used by subsequent NLP tasks, the response speed of the ENER API must be fast enough for the subsequent tasks to achieve a high speed. Consequently, we executed the ENER API with the test dataset (containing 2869 sentences) and evaluated the response time of the API. The average response time of a sentence (a query) is 195 ms (0.195 second). This response speed is fast enough for various tasks such as generating answer for an intelligent chatbot or a search engine session. Figure 5 shows the relation between the response time and the length of the input sentence (calculated by the number of tokens, each token is a word produced by the morphological analyzer). When the input sentence length increases, the response time increases nearly linearly (except when the sentence is too long, as we have a small number of such sentences so the variance is large). The typical sentence length in Japanese is from 10 to 20 tokens so the speed of the ENER is fast in most cases.

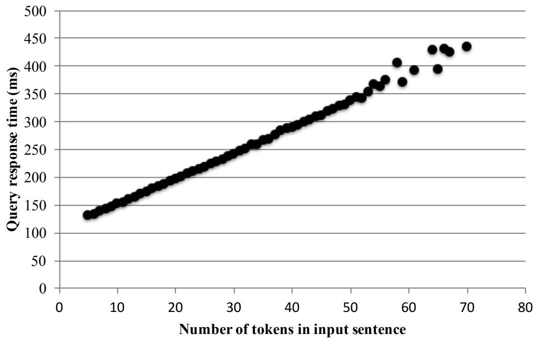


Figure 5: Relation between input sentence length and response time of the API

4 Application of the ENER API

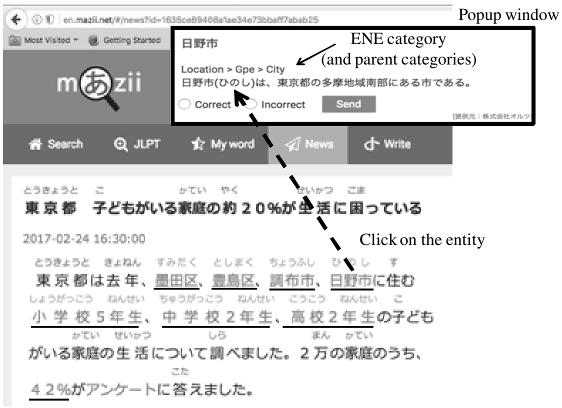
In this section, we present a real-world application of the AL+ ENER API: glossary linking in an online news service.

4.1 Mazii: an online news service for Japanese learners

The Mazii News service[4] is an online news service for Japanese learners. For each sentence in a news article, Mazii automatically analyzes it and creates a link for each word that it recognizes as an ENE or an entry in its dictionary. This will help Japanese learners to quickly reference to the words/entities when they do not understand the meaning of the words/entities. To recognize ENEs in a news article, Mazii inputs each sentence of the article into the AL+ ENER API (sentence boundary detection in Japanese is very simple because Japanese language has a special symbol for sentence boundary mark). Because the AL+ ENER API also returns the position (and the length) of the ENEs, Mazii can easily create a link to underline the ENEs in the sentence. When a user clicks on a link, Mazii will open a popup window to provide details information concerning the entity: the ENE category (with parent categories) of the entity, the definition of the entity (if any). Figure 6

[4] http://en.mazii.net/#/news

shows a screenshot of the Mazii ENE linking results.



日野市

Location > Gpe > City

日野市(ひのし)は、東京都の多摩地域南部にある市である。

東京都　子どもがいる家庭の約２０%が生活に困っている

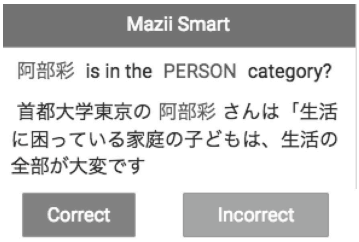
2017-02-24 16:30:00

東京都は去年、墨田区、豊島区、調布市、日野市に住む小学校５年生、中学校２年生、高校２年生の子どもがいる家庭の生活について調べました。２万の家庭のうち、４２%がアンケートに答えました。

Figure 6: Mazii entity linking with AL+ ENER API, the underlined entities are linked. When a user clicks on a link (as shown in the Figure, a mention to a city in Japan is clicked), a popup window will open and show the ENE category hierarchy of the corresponding ENE.

4.2 Collecting user feedbacks

Mazii has more than 4 thousands daily active users and many users click on the linked ENEs. This provides us a big chance to obtain user feedbacks about the prediction results of the AL+ ENER API. We have implemented two interfaces to collect user feedbacks, as shown in Figure 6 and Figure 7.



阿部彩 is in the PERSON category?

首都大学東京の 阿部彩 さんは「生活に困っている家庭の子どもは、生活の全部が大変です

Figure 7: Collecting ENE user feedback from Mazii with playcard game

In Figure 6, when a user clicks on an entity, we display the ENE hierarchy of the entity in a popup window. We also display two radio buttons: **Correct** and **Incorrect** to let the user give us feedbacks. If the user chooses **Incorrect** then we also ask the user the correct category of the entity.

Using the method in Figure 6, we can only collect feedbacks when the users click on the entities. However, the number of clicks is often much smaller than the number of views. To increase the user feedbacks, we invented a playcard game for language learners, as shown in Figure 7. When a user views an article, we show a frame with a question asking about the correct category of an ENE in the article (we also provide the sentence which includes the ENE to gather the context for the CRF-SVM and RNN models). If the user reacts to this frame (by pressing Correct/Incorrect button), we store the feedback and move to the next ENE in our database. This involves the user in a language learning game and helps he/she to study many new words as well as grammatical constructs.

4.3 User feedback statistics

In this section, we show some statistics that we derived from the user feedback log of the Mazii News service. We collected the user feedback log (including the view, click and correct log) in 3 months (from Dec 2016 to Feb 2017). We then count the number of views, clicks and number of feedbacks (number of times the Correct/Incorrect button is pressed) and number of Correct times for each ENE categories. We calculate the correct ratio (%Correct) by the number of corrects divided by number of feedbacks (Correct/Feedback).

Category	View	Click	Feedback	%Correct
Date	360,625	7,100	1,421	95.50
N_Person	139,191	1,934	523	98.47
Province	109,974	9,880	439	94.76
...	...	...	...	...
Animal Part	6,514	637	8	100.00
Broadcast Program	6,121	1,003	21	47.62
Clothing	4,079	632	14	85.71
...	...	...	...	...
Fish	656	474	2	100.00
Fungus	615	106	1	0.00
Religion	614	227	4	100.00
Total	1,582,081	138,404	5,198	88.96

Table 3: Number views, clicks, feedbacks and percentage of correct times from the Mazii feedback log

Table 3 shows the experiment results. The correct ratio (%Correct) is 88.96% on 96 categories with more than 100 views and have at least one user feedback. The table also shows the detailed numbers for some categories, sorted by number of views. The average click-through-rate (CTR=Click/View) is 8.7%, which is very high compared to the average CTR of display ads (about 0.4%) (Zhang et al., 2014). This proves that the users are interested in the linked ENEs. Moreover, the percentage of correct times shows that the ENER API is good enough to provide useful information to the users.

5 Related Work

The ENE hierarchy that we recognize in this paper is proposed in (Sekine et al., 2002). (Sekine and Nobata, 2004) proposed a Japanese rule-based ENER with a precision of 72% and recall of 80%. The performance of the rule-based ENER is good if the ENEs containing in the text are included in the dictionary or the rules can capture the patterns in which the ENEs appeared. However, ENEs often evolve with time, new ENEs are frequently added and their meaning might be changed. Consequently, rule-based systems might not work well after a several years. In the presented system, we re-use the rules and dictionary in (Sekine and Nobata, 2004) but we also add machine learning models to capture the evolution of the ENEs. The proposed model can be retrained at anytime if we have new training data. Iwakura et al. (Iwakura et al., 2011) proposed an ENER based on decomposition/concatenation of word chunks. They evaluated the system with 191 ENE categories and achieved an F-score of 81%. However, in their evaluation, they did not evaluate directly on input sentences, but only on correct chunks. Moreover, they did not deal with word boundaries as stated in their paper. Therefore, we cannot compare our results with theirs.

6 Conclusion

We presented an API for recognition of Extended Named Entities (ENEs). The API takes a sentence as input and outputs a JSON containing a list of ENEs with their categories. The API can recognize named entities at deep level with high accuracy in a timely manner, and has been applied in real-life applications. We described an application of the ENER API to a Japanese online news service. The experimental results showed that the API achieves good performance and is fast enough for practical applications.

Acknowledgments

We would like to thank Yoshikazu Nishimura, Hideyuki Shibuki, Dr. Phuong Le-Hong and Maya Ando for their precious comments and suggestions on this work.

References

Patrice Bellot, Eric Crestan, Marc El-Bèze, Laurent Gillard, and Claude de Loupy. 2002. Coupling Named Entity Recognition, Vector-Space Model and Knowledge Bases for TREC 11 Question Answering Track. In *Proc. of TREC 2002*.

Jenny Rose Finkel, Trond Grenager, and Christopher D. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In *Proc. of ACL 2005*. pages 363–370.

Tomoya Iwakura, Hiroya Takamura, and Manabu Okumura. 2011. A Named Entity Recognition Method based on Decomposition and Concatenation of Word Chunks. In *Proc. of IJCNLP 2011*. pages 828–836.

Andrew McCallum and Wei Li. 2003. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In *Proc. of CoNLL 2003*. pages 188–191.

David Nadeau and Satoshi Sekine. 2007. A Survey of Named Entity Recognition and Classification. *Linguisticae Investigationes* 30(1):3–26.

Ryohei Sasano and Sadao Kurohashi. 2008. Japanese Named Entity Recognition Using Structural Natural Language Processing. In *Proc. of IJCNLP 2008*. pages 607–612.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *Proc. of LREC 2004*. pages 1977–1980.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended Named Entity Hierarchy. In *Proc. of LREC 2002*. pages 1818–1824.

Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. 2016. Fine-Grained Named Entity Classification with Wikipedia Article Vectors. In *Proc. of Int'l Conf. on Web Intelligence (WI 2016)*. pages 483–486.

Koichi Takeuchi and Nigel Collier. 2002. Use of Support Vector Machines in Extended Named Entity Recognition. In *Proc. of CoNLL 2002*.

Hiroyasu Yamada, Taku Kudo, and Yuji Matsumoto. 2002. Japanese Named Entity Extraction Using Support Vector Machine. *Transactions of Information Processing Society of Japan (IPSJ)* 43(1):44–53.

Weinan Zhang, Shuai Yuan, and Jun Wang. 2014. Optimal Real-Time Bidding for Display Advertising. In *Proc. of KDD 2014*. pages 1077–1086.

Guodong Zhou and Jian Su. 2002. Named Entity Recognition Using an HMM-Based Chunk Tagger. In *Proc. of ACL 2002*. pages 473–480.