An Empirical Study on Fine-Grained Named Entity Recognition

Khai Mai[1,*] **Thai-Hoang Pham**[1,*] **Nguyen Minh Trung**[1] **Nguyen Tuan Duc**[1]
Danushka Bolegala[2] **Ryohei Sasano**[3] **Satoshi Sekine**[4]
[1)] Alt Inc [2)] University of Liverpool
[3)] Nagoya University [4)] Riken AIP
{mai.tien.khai, pham.thai.hoang, nguyen.minh.trung, nguyen.tuan.duc}@alt.ai,
danushka.bollegala@liverpool.ac.uk, sasano@i.nagoya-u.ac.jp, satoshi.sekine@riken.jp

Abstract

Named entity recognition (NER) has attracted a substantial amount of research. Recently, several neural network-based models have been proposed and achieved high performance. However, there is little research on fine-grained NER (FG-NER), in which hundreds of named entity categories must be recognized, especially for non-English languages. It is still an open question whether there is a model that is robust across various settings or the proper model varies depending on the language, the number of named entity categories, and the size of training datasets. This paper first presents an empirical comparison of FG-NER models for English and Japanese and demonstrates that LSTM+CNN+CRF (Ma and Hovy, 2016), one of the state-of-the-art methods for English NER, also works well for English FG-NER but does not work well for Japanese, a language that has a large number of character types. To tackle this problem, we propose a method to improve the neural network-based Japanese FG-NER performance by removing the CNN layer and utilizing dictionary and category embeddings. Experimental results show that the proposed method improves Japanese FG-NER F-score from 66.76% to 75.18%.

1 Introduction

Named entity recognition (NER) is a well studied topic in natural language processing. There have been many methods proposed for NER, including the conventional methods based on Conditional Random Fields (CRF) (McCallum and Li, 2003), Support Vector Machines (SVM) (Yamada et al., 2002; Takeuchi and Collier, 2002) and Hidden Markov Model (HMM) (Zhou and Su, 2002). Recently, neural network based methods, such as LSTM+CNN+CRF (Ma and Hovy, 2016) or BiLSTM/LSTM-CRF/Stack LSTMs (Lample et al., 2016; Misawa et al., 2017), have achieved state-of-the-art performance. However, while most existing studies mainly focus on recognizing a relatively small number of named entity (NE) categories (e.g., ten or twelve categories) such as Person, Organization, Location, Artifact, etc., modern NLP applications often require domain-specific and fine-grained (FG) NER with hundreds of NE categories. For example, a movie recommendation system might require the recognition of movie names, but does not need to recognize Locations. Similarly, a chatbot software might require not only the recognition of Organization, but also the fine-grained classification to recognize a music band name to answer the question "Which band was Paul in", from the information shown in Figure 1.

A fine-grained named entity recognition (FG-NER) model refers to a NER model that can recognize and classify a large number of entity categories (e.g., hundreds of NE categories). In classical coarse-grained named entity (NE) definition, often less than ten named entity categories are defined. For example, in the CoNLL-2003 Named Entity Recognition task, there are four NE categories: Person, Location, Organization and Miscellaneous (Sang and Meulder, 2003). Ritter et al. (2011) proposed a NER algorithm to recognize ten categories of entities from Twitter text. On the other hand, in FG-NER, there are hundreds of NE categories, which are fine-grained classification of coarse-grained categories.

*) Equally contributed to the paper

Proceedings of the 27th International Conference on Computational Linguistics, pages 711–722
Santa Fe, New Mexico, USA, August 20-26, 2018.

Paul, a former member of The Beatles, known for "Let It Be",
Person *Organization* *Artifact*

will be holding a concert at Carnegie Hall in New York.
Location *Location*

(a) Named entity recognition result

Paul, a former member of The Beatles, known for "Let It Be",
Person *Org > Show_Org* *Product > Art > Music*

will be holding a concert at Carnegie Hall in New York.
Facilty > GOE > Theatre *Location > GPE > City*

(b) Fine-grained NER result

Figure 1: Example of NER and FG-NER

For example, Sekine (2008) divided the coarse-grained category Organization into the fine-grained categories such as Political_Party, Military, Sports_Organization, Show_Organization, as shown in Figure 2.

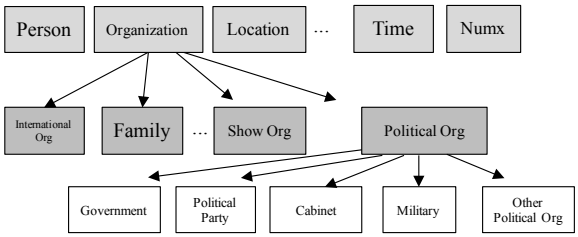


Figure 2: Sekine's Extended Named Entity (ENE) hierarchy

FG-NER is still an open research domain, with little information concerning the state-of-the-art performance, the relation between training data size and performance, and how to select the best model for different settings of training data size and target language. In FG-NER, because the number of NE categories is large, some categories might face with the data sparseness problem, whereas some other categories might have a large number of training samples in a dataset. Hence, it would be worth investigating the relation between dataset size and the performance of the system. The current state-of-the-art method for English NER (coarse-grained NER) is a neural network-based method, which uses convolutional neural network (CNN) to calculate the character level embeddings (Ma and Hovy, 2016). This leads to the question whether this method works well for languages with a large number of character types, such as Japanese.

In this paper, we first investigate the relationship between the F-score of various FG-NER algorithms with the size of training datasets for both English and Japanese. Second, we suggest the direction to choose an appropriate FG-NER algorithm for appropriate target language and training data size. We show that the state-of-the-art method for English NER also performs well for English FG-NER. On the other hand, for Japanese FG-NER, the state-of-the-art method does not work well. To solve this problem, we propose a method to significantly improve the neural network-based Japanese FG-NER performance by removing the CNN layer, as well as utilizing dictionary and category embeddings information. Experiments show that, the proposed method improves the F-score of the Japanese FG-NER system from 66.76% to 75.18%, which is a wide margin. We applied the proposed method to build an FG-NER system that can recognize 200 categories of named entities in the Sekine's Extended Named Entity Hierarchy (ENEH) (Sekine, 2008). To the best of our knowledge, the proposed system achieves the state-of-the-art performance in the task of recognizing 200 NE categories in the Sekine's ENEH. We publish the test dataset[1] and our FG-NER API on the Web to allow other researchers to freely use[2].

The rest of this paper is organized as follows. In Section 2, we describe the fine-grained named entity tag sets and datasets. We present various algorithms for FG-NER and show our proposal to achieve high performance for Japanese FG-NER in Section 3. We present detailed experimental results to find out the relationships between models, training data size and target languages in Section 4. Finally, Section 5 concludes the paper.

[1] https://fgner.alt.ai/duc/ene/testsets/comp/en/
[2] https://fgner.alt.ai/extractor/

2 Fine-Grained Named Entity Tag Sets and Datasets

2.1 FG-NER tag set

The first challenge in FG-NER is defining a comprehensive tag set with a very large number of entity categories (Ling and Weld, 2012). There are two methods for defining a tag set (i.e., set of entity categories to recognize) in previous studies on FG-NER. The first method is to take the entity categories from a knowledge base such as Freebase (Bollacker et al., 2008) or YAGO (Suchanek et al., 2007), filtering out the categories with a small number of entities and merging the categories with similar semantic meaning into one FG-NER category (Ling and Weld, 2012; Yosef et al., 2012; Gillick et al., 2014). The second method is to manually build an entity hierarchy to cover important domains in the real world. Following the second method, Sekine et al. proposed an Extended Named Entity Hierarchy (ENEH), which contains 200 entity categories in a three-layer hierarchy, as shown in Figure 2 (Sekine et al., 2002; Sekine, 2008).

In this paper, we use the entity hierarchy described by Sekine (2008), which contains 200 NE categories at the leaf-level, as our tag set[3]. At the top level of the hierarchy, there are about twenty coarse-grained named entity categories, such as Person, Organization, Location, Facility, Product, Event, ...Each top-level categories is further divided into several second-level categories as shown in Figure 2. Each second-level category is in turn divided into several leaf-level categories.

We use this hierarchy because it is carefully designed by humans, it does not ignore important domains with small numbers of entities (e.g., Continental_Region) and it includes a systematic categorization of date/time and number. Moreover, for subsequent applications such as search engines or chatbot platforms to easily utilize the FG-NER results, we want to classify an entity to exactly one category in the hierarchy based on the context in which the entity appears. Consequently, we need an entity category hierarchy that does not allow overlap between the categories (Sekine, 2008). For all experiments in this paper, we will build Fine-grained NERs to recognize the leaf-level categories in this hierarchy. The parent level categories can be easily inferred once we recognize the leaf-level categories.

2.2 FG-NER Dataset

We hired several human annotators to annotate two text corpora to create FG-NER tagged datasets for English and Japanese. The number of annotators for English is ten and the number of annotators for Japanese is seven. All of the annotators are native speakers of the corresponding language.

For each category in the Sekine's Extended Named Entity (ENE) Hierarchy (Sekine, 2008), the human annotators are first asked to write down 100 entities that belong to each ENE category. We then search the Web for sentences that contain these entities. In the search result, we retrieve the sentences that include at least one entity of the corresponding category. The human annotators then tag the sentences with 200 ENE labels. For example, for the entity "Tokyo", the Web search results might contain the sentence "Tokyo is the capital of Japan". This sentence must then be tagged as "<City>Tokyo</City> is the capital of <Country>Japan</Country>". Consequently, we have 20,000 sentences for English and 20,000 sentences for Japanese (as the number of leaf-level categories defined in Sekine's ENEH is 200). Note that the number of entities is larger than the number of sentences because in one sentence we might have multiple entities, of different NE categories.

After filtering out erroneous sentences (sentences with invalid tag format, e.g., without closing the tag </City>), we obtain totally 19,800 well-formed English sentences and 19,594 well-formed Japanese sentences. Each category has at least 100 sentences containing the entities of that category. We divided the dataset into three subsets: the training set (70% of the total data), development set (10%) and test set (20%), as shown in Table 1. We use the development set to check the stop condition while training our LSTM model.

To have an estimation of the difficulty of the FG-NER annotation task, we measured the coherence between the annotators by calculating two coefficients for some categories in the English dataset: the Fleiss' kappa (κ) and the F-score of an annotator when assuming that another annotator created gold-

[3]The full tag set is here: https://nlp.cs.nyu.edu/ene/version7_1_0Beng.html

Table 1: Statistics of the datasets

Dataset	English		Japanese	
	Sents	Entities	Sents	Entities
Train	13,860	27,107	13,749	37,128
Dev	1,980	3,870	1,948	5,304
Test	3,960	7,739	3,897	10,485

Table 2: Agreement between annotators

Category	Fleiss' κ	F-score
Country	0.977	0.966
Dish	0.890	0.738
Car_Stop	0.873	0.672
Organization_Other	0.835	0.483

standard data[4]. The results are shown in Table 2. We choose these categories to calculate the coefficients because they represent typical categories in the dataset: a category with limited number of frequently used entities (*Country*), a category with entities that are often not proper nouns (*Dish*), a category with ambiguous and complicated location names (*Car_Stop*) and a category that is ambiguously defined (*Organization_Other*).

The κ coefficient is calculated on tokens so it is only slightly different between the four categories in Table 2. This is because the ratio between the number of tokens with label "Other" is very large, compared against the number of tokens with specific NE category labels. We calculated Fleiss' κ on tokens but not on entities because at entity level, there are some cases in which two annotators made overlapping tags, but not identical. For example, Annotator1 might tag "<City>Greater Tokyo Area</City>" and Annotator2 might tag "Greater <City>Tokyo</City> Area". If we calculate on entity level then the score would be 0 but if we calculate on token level, the score is greater than 0. Consequently, we calculate the κ based on B/I/O tags at token level.

On the other hand, when calculating the entity-based F-score, the difference is very large between the category *Country* and *Organization_Other*. This is because the category *Country* is very easy to recognize, as there are only about 200 entities frequently used in this category, whereas, recognizing *Organization_Other* or *Car_Stop* is very difficult because of the ambiguity. This also indicates that the performance of an FG-NER system tends to depend on the categories and we can confirm this in the experimental results in the next sections.

3 Fine-Grained Named Entity Recognition Methods

3.1 Dictionary and Rule-based FG-NER

The simplest method for FG-NER is using a dictionary and a set of rules. Sekine and Nobata (2004) presented a dictionary and rule-based Japanese FG-NER system that contains more than 1400 rules to recognize 140 entity categories.

In this work, we added 200 rules to the existing 1400 rules by Sekine and Nobata to create a rule set of 1600 rules to classify 200 NE categories in the Sekine's Extended Named Entity Hierarchy. We then built a rule-based Japanese FG-NER model to recognize 200 NE categories based on these 1600 rules.

We use a Japanese FG-NER dictionary containing 1.6 million Wikipedia entities in this model. In the 1.6 million entities in the dictionary, only 70 thousand entities are assigned NE tags by human, the rest are assigned by an existing Wikipedia NE labeling algorithm (Suzuki et al., 2016), which gives a score for each (entity, NE category) pair. We created similar rules for English FG-NER and we translated the Japanese dictionary into English by looking up parallel entries in Wikipedia.

We use this method as a baseline for performance evaluation of FG-NER systems.

3.2 CRF+SVM hierarchical classifier for FG-NER

Hierarchical classifiers have been successfully used in previous research for FG-NER (Ling and Weld, 2012; Yosef et al., 2012). Ling and Weld proposed FIGER, an FG-NER system with the entity categories taken from Freebase tags (Ling and Weld, 2012). In FIGER, the entity category is represented as a path, such as `/location/city` or `organization/company`. FIGER divides the entity categories into

[4]In this case, we only have two annotators for each categories, so Fleiss' kappa is equal to Cohen's kappa

a hierarchy of two layers: the categories in the first layer corresponding to the categories in coarse-grained NER systems, whereas, the second layer (the leaf-layer) contains fine-grained entity categories. FIGER uses a hierarchical classifier that contains a CRF at the top layer for sequence labelling and then a Perceptron at the second layer for classification of the entities into fine-grained categories.

In this work, we propose a hierarchical classification method in which CRF is used for sequence labelling at top-level and SVM is used for named entity classification at leaf-level of the Sekine's ENE Hierarchy, as shown in Figure 3. We use SVM at the leaf-level to classify an entity to fine-grained categories because SVM is good for classification tasks.

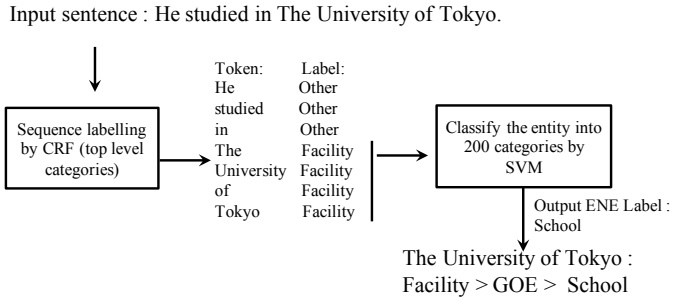


Figure 3: CRF+SVM based hierarchical classifier for FG-NER

Specifically, we use a training dataset (containing FG-NER tagged sentences) to train a CRF model to tag the input sentences with the top-level ENE categories in Figure 2. As illustrated in Figure 2, at the top level, we have only less than 20 categories of entities, thus using a CRF model here would achieve comparable performance with existing NER systems. We cannot directly use CRF for a large number of categories at leaf-level of the hierarchy, because with this number of classes, it would take a huge amount of memory and running time to train the CRF model. Actually, we have tried to use CRF for 200 classes, but the training process took a long time and did not finish. After tagging the sentences with the top-level categories, we can convert the FG-NER problem into a simple classification problem (not a sequence labeling problem anymore), thus we can use SVM to classify the extracted entities 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 entities into the leaf-level categories.

We use the following features for both SVM and CRF: 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 Romanji), is capital letter, and the last 2-3 characters. Those features are proved to be useful in previous work on named entity recognition (Ling and Weld, 2012; Yosef et al., 2012; Yogatama et al., 2015; Suzuki et al., 2016).

Once we have the sequence labelling result, we have already known the surfaces and the top-level categories of the entities in the input sentence. We then use SVM to classify the entities into leaf-level categories. Because the number of leaf-level categories in each top-level categories is also not too large (e.g., less than 15), SVM can achieve a reasonable performance at this step.

We also propose a method to incorporate dictionary information in both CRF and SVM step to improve the entire performance, as described in Section 3.4.

3.3 LSTM+CNN+CRF model for FG-NER

We re-implemented the LSTM+CNN+CRF NER model described by Ma and Hovy (2016) and adjust the model to work with FG-NER. The LSTM+CNN+CRF model originally described by (Ma and Hovy, 2016) is for NER problem with few NE categories. It first uses Convolutional Neural Network (CNN) to learn character level embeddings in the training process. For NLP tasks, previous works have shown that CNN is likely to extract morphological features such as prefix and suffix effectively (Ma and Hovy,

2016; dos Santos and Guimarães, 2015; Chiu and Nichols, 2016). The model then concatenates the character level embeddings with word embeddings to create a feature vector for each token in the input sentence. The input sentence is then fed to a BiLSTM network (Bi-directional Long-Short Term Memory network). Finally, CRF is used at the top layer of the BiLSTM to explore the correlations between outputs and jointly decode the best sequence of labels (i.e., NE categories).

For both English and Japanese FG-NER task, we use pre-trained word embeddings as input for our models. Previous studies have shown that GloVe achieves the best performance for English NER task (Reimers and Gurevych, 2017). Consequently, we use the embeddings based on GloVe for English[5]. For Japanese, we use pretrained word2vec[6] embeddings. The vector dimension is 300 for English and 200 for Japanese.

We use the default hyperparameters by Ma and Hovy (2016) in our model: $learning\_rate = 0.01$, $batch\_size = 10$ and $decay\_rate = 0.09$.

3.4 Incorporating dictionary information

Dictionary information (gazetteer feature) has been proved to be efficient in many NER and FG-NER tasks (McCallum and Li, 2003; Sekine and Nobata, 2004; Yosef et al., 2012). While there are previous studies that use dictionary for CRF (McCallum and Li, 2003) or SVM (Yosef et al., 2012) in the NER/FG-NER tasks, we believe that dictionary information would be useful in both sequence labelling and entity category disambiguation phase in the CRF+SVM method. Furthermore, the dictionary information can also be used in LSTM+CNN+CRF method. Consequently, we propose a method that efficiently utilizes dictionary information in the method LSTM+CNN+CRF and in both sequence labelling (CRF) and entity category disambiguation (SVM) phase of the method CRF+SVM.

We search the dictionary and assign a label in the set $\{B, I, O\}$ for each token in the input sentence $w_1 w_2 \dots w_n$, in which a token w_i is assigned the label *B* if it is a start token of an entity in the dictionary, label *I* if it is a token inside an entity in the dictionary, otherwise, it is assigned the label *O*. If there is a conflict (e.g., overlapping with two different entities in the dictionary) then we take the entities with the largest number of tokens. This is because we want to tag the longest sequence that could be an entity (e.g., if we have "United States" and "United States Army" then we take "United States Army"). We do not directly use these labels as the final results of the sequence labelling phase since they are not reliable as we drop all context information while assigning labels. Instead, we use these labels as features for CRF model for sequence labelling the entire input sentence or we add it to the additional dimensions of the vector representation of a token in the method LSTM+CNN+CRF.

Other than features from previous research on entity disambiguation (Yosef et al., 2012; Suzuki et al., 2016), we propose to use the following feature derived from the dictionary. We tokenize all the entities in the dictionary and calculate the probability that a token w is contained in an entity of type c :

$$P(c|w) = \frac{\text{count}(w, c)}{\text{count}(w, *)} \tag{1}$$

in which $\text{count}(w, c)$ is the number of occurences of w in an entity of type c, whereas, $\text{count}(w, *)$ is the total number of times that w appears. We then use this probability as a feature for SVM to classify the entities into leaf-level categories in the method CRF+SVM and we directly add this feature to the feature vector of a token in the LSTM+CNN+CRF method. This feature is helpful because it represents the likelihood that an entity e (which is a sequence of tokens $w_i w_{i+1} \dots w_{i+m}$) is contained in the category c.

We add these features to the CRF+SVM method to have the CRF+SVM+Dict method. Similarity, while using these features for LSTM+CNN+CRF, we obtained LSTM+CNN+CRF+Dict method.

3.5 Utilize entity category embeddings to improve Japanese FG-NER performance

For languages with a large number of character types such as Japanese, the CNN layer in the LSTM+CNN+CRF+Dict network architecture might not work well. This is because with a large

[5]https://nlp.stanford.edu/projects/glove/

[6]http://www.cl.ecei.tohoku.ac.jp/∼m-suzuki/jawiki_vector/

number of character types (e.g., 2000 frequently used characters in Japanese), it is difficult to calculate high quality character embeddings. Consequently, we propose a modification to the existing LSTM+CNN+CRF network architecture to alleviate this problem: We remove the CNN layer from the LSTM+CNN+CRF+Dict method to have the LSTM+CRF+Dict method. Moreover, we add another additional information concerning the entity category, as described below.

In the spirit of taking advantage of available dictionaries to improve our model as in CRF+SVM, we propose to integrate statistical information from the dictionary to improve the word embeddings. Specifically, for a word, we form a vector in which each element is the probability that this word is contained in an entity category, as follows:

$$C(w) = (x_1, x_2, \ldots, x_{n-1}, x_n) \tag{2}$$

where x_i is the probability that word w is contained in i^{th} category (c_i) :

$$x_i = P(c_i|w) = \frac{\text{count}(w, c_i)}{\text{count}(w, *)} \tag{3}$$

We call these "entity category embeddings". For new words or words that do not appear in any entity category in the dictionary, their entity embeddings are zero at all dimensions. We concatenate this entity category embedding vector with the original word embedding vector to form a new word embedding vector. We call the **LSTM+CRF+Dict** method that also utilizes entity category embeddings as **LSTM+CRF+Dict+Cate**.

4 Experiments

4.1 Performance comparison between FG-NER methods

We evaluated the performance of the FG-NER methods using the test dataset, as described in Table 1. We calculated the Precision, Recall and F1-score for each category and the micro-average Precision and Recall over all 200 entity categories. Finally, we derived the average F1-score from the average Precision and Recall.

Method	English F-score (%)	Japanese F-score (%)
Rule+Dict	24.43	48.29
FIGER(Ling and Weld, 2012)	23.41	-
CRF+SVM+Dict	72.58	**73.30**
LSTM+CNN+CRF (Ma and Hovy, 2016)	80.93	66.76
LSTM+CNN+CRF+Dict	**83.14**	70.34
LSTM+CRF+Dict	81.89	73.05

(a) Average F-score of the FG-NER methods

Method	English	Japanese
LSTM+CNN+CRF (Ma and Hovy, 2016)	80.93	66.76
LSTM+CNN+CRF+Dict	**83.14**	70.34
LSTM+CNN+CRF+Dict+Cate	82.29	-
LSTM+CRF+Dict	-	73.05
LSTM+CRF+Dict+Cate	-	**75.18**

(b) Average F-score of the FG-NER systems with and without entity category embeddings

Category	Precision (%)	Recall (%)	F-score (%)
URL	100.00	100.00	100.00
Phone_Number	100.00	100.0	100.00
Cabinet	95.24	100.00	97.56
Country	90.06	92.67	91.35
...	...	...	...
Award	80.65	92.59	86.21
Car_Stop	80.00	76.19	78.05
Book	87.50	63.64	73.67
Dish	67.31	76.09	71.43
...	...	...	...
Mollusk_Arthropod	65.79	64.10	64.93
Art_Other	63.16	57.14	59.99
Organization_Other	57.14	48.00	52.17
Average	**83.25**	**83.04**	**83.14**

(c) Precision, Recall, F-score of the method LSTM+CNN+CRF+Dict on the English test dataset, sorted by F-score

Table 3: Performance of FG-NER methods

The evaluation results are shown in Table 3a. Rule and dictionary-based methods (**Rule+Dict**) can only achieve an F-score of less than 50%, even we used more than 1000 rules and a huge number of entries in the dictionaries. Because there are many Wikipedia entries that are in Japanese Wikipedia but they are not in English Wikipedia and we translated the dictionary from Japanese by looking up parallel

entries in English Wikipedia, these entities do not appear in the English dictionary. Consequently, the F-score of English **Rule+Dict** is very low.

The method **FIGER** is the CRF+Perceptron hierarchical classifier described by Ling and Weld (Ling and Weld, 2012). We use the published source code of **FIGER**[7] and our training dataset (identical to the training dataset of other methods in this paper) to get an FG-NER model. We evaluated this model with the test dataset described in the previous section. We can only evaluate **FIGER** with English because the source code of **FIGER** can only work with English (e.g., the feature extraction code can only work with English if we don't modify it). Although the architecture of the method **FIGER** is very similar to that of **CRF+SVM+Dict**, the performance is very different (the F-score is 24.43%, compared to 72.58%). This is because of several reasons. First, **FIGER** is designed for classifying the two-layer hierarchy of named entities based on Freebase. This hierarchy is different from the three-layer hierarchy of Sekine's Extended Named Entity Hierarchy (ENEH) that we use in this work. Second, we use SVM at the leaf-level, instead of Perceptron. We believe that SVM gives high performance than Perceptron in multi-class classification tasks. And finally, the training data size is not large enough for **FIGER** to work well. **FIGER** uses millions of training samples (which are automatically extracted from Wikipedia) to train the system (Ling and Weld, 2012), whereas, in our method, we only have about 13 thousand sentences in the training dataset. Automatically creating the NE tagged data is a reasonable approach for FG-NER definitions that are derived from existing knowledge bases like Freebase or Wikipedia. For FG-NER definitions that are not based on existing knowledge bases, it is difficult to automatically extract the NE tagged data from a large text corpus like Wikipedia. Consequently, for FG-NER system working with these definitions, we must design an algorithm that requires only a small amount of training data.

We can observe that the state-of-the-art method for English NER (Ma and Hovy, 2016) also works well with English FG-NER, as **LSTM+CNN+CRF** outperforms **CRF+SVM+Dict** by a wide margin (80.93%, compared to 72.58%). However, for Japanese FG-NER, the situation is different. The method **CRF+SVM+Dict** achieves the best performance, and it significantly outperforms the original **LSTM+CNN+CRF** model in (Ma and Hovy, 2016). Even when we add dictionary information, **CRF+SVM+Dict** still outperforms **LSTM+CNN+CRF+Dict** for Japanese FG-NER (the F-score is 73.30 and 70.34, respectively). We measured the statistical significance of the F-score difference by an approximate randomization test (Chinchor, 1992) with significance level $\alpha = 0.05$ and 99,999 iterations. The randomization test is used because we want to check the difference between the micro-average F-score over all ENE categories. The result of this test indicates that the difference in F-score is statistically significant (the p-value is 10^{-5}). The reason for this difference lies in the character level embeddings that **LSTM+CNN+CRF+Dict** creates in the CNN layer. For English, we only have less than 30 characters in the alphabet, whereas, in Japanese, we have more than 2,000 Hiragana, Katakana and Kanji characters that are frequently used (the total number of Japanese characters is above 10,000). Because the number of characters is large in Japanese, the CNN layer does not work well to create good quality character embeddings.

We verified the above reason by the following experiment: we removed the CNN layer from LSTM+CNN+CRF+Dict and we measured the performance on the test dataset. Removing the CNN layer means disabling the character level embedding features in the LSTM+CNN+CRF+Dict method. The result of this experiment is shown in the last row of Table 3a. We can observe that, removing the CNN layer boosts the performance of the neural network based method from 70.30% (**LSTM+CNN+CRF+Dict**) to 73.05% (**LSTM+CRF+Dict**), making it comparable with the performance of **CRF+SVM+Dict** (73.30%) for Japanese FG-NER. However, for English FG-NER, the performance decreased if we remove the CNN layer. This proves that the CNN layer works well for English, but it does not work for Japanese.

We also verify the difference between **CRF+SVM+Dict** and **LSTM+CRF+Dict** using a randomization test as described previously. For Japanese, the difference between **CRF+SVM+Dict** and **LSTM+CRF+Dict** is not statistically significant. This suggests that with enough amount of training data and dictionary information, we can use both CRF or LSTM for FG-NER problem.

[7] https://github.com/xiaoling/figer

4.2 Result of removing CNN and using entity category embeddings

To verify the effectiveness the proposed method for Japanese FG-NER, we compared the performance of the best neural network-based model for each language with and without the entity category embeddings. The results are shown in Table 3b.

Because removing the CNN layer does not improve the performance of English FG-NER, we compared **LSTM+CNN+CRF** with the method **LSTM+CNN+CRF+Dict** and **LSTM+CNN+CRF+Dict+Cate** for English. We observe that, by adding dictionary information, we can improve the F-score from 80.93% to 83.14%. This improvement is statistically significant under a randomization test (similar to the tests in Section 4.1). However, if we further add the entity category embedding information (in **LSTM+CNN+CRF+Dict+Cate**), the performance slightly decreased (from 83.14% to 82.29%, which is statistically significant under the randomization test). This is because the quality of the translated English dictionary is not good enough to have practical statistics information.

Table 3b also shows a series of improvements we made for Japanese FG-NER. First, if we add the dictionary information, we obtained a similar result with that of English FG-NER (we boost the performance from 66.76% to 70.34%, as shown in the 1st and 2nd row of Table 3b). Second, if we remove the CNN layer from the network, we improve the performance from 70.34% to 73.05%, as described in the previous section. Finally, if we add category entity embeddings, we further improve the performance from 73.05% to 75.18%. This makes the method **LSTM+CRF+Dict+Cate** significantly outperforms the method **CRF+SVM+Dict** (the randomization test result shows that the improvement is statistically significant). This proves the effectiveness of the newly added entity category embedding feature for Japanese FG-NER.

With this result, we can unify the FG-NER model for both English and Japanese: we don't need to use **CRF+SVM** anymore, but we can utilize neural network-based models for both languages. We can have an identical system for both English and Japanese FG-NER by simply setting an option to enable/disable the CNN layer and the entity category embedding information to switch between English and Japanese. This makes the engineering of the system easier and helps to cut the maintenance cost of the FG-NER system.

4.3 Performance of each category

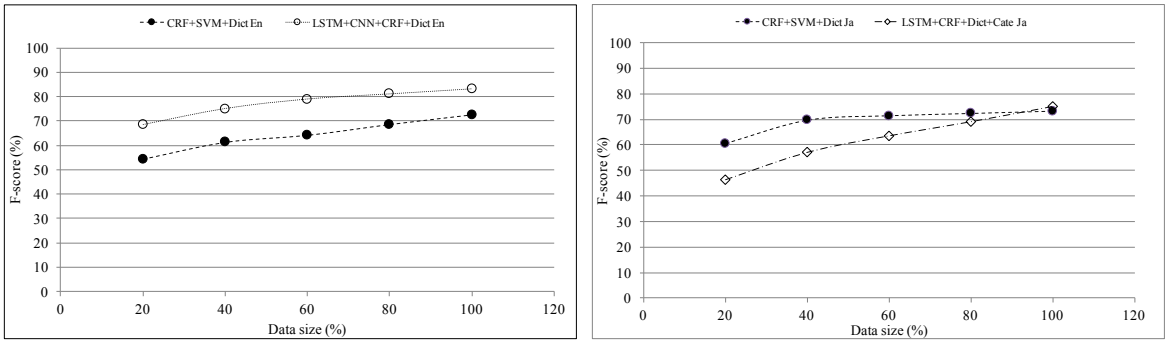
In the 200 NE categories in the Sekine's ENEH (Sekine, 2008), there are several categories that can be recognized by rule-based method (such as URL or Email). There are also some categories that are actually not named entities, and difficult to recognize, such as Mollusk_Arthropod (cellar spider, turbinidae,...) or Art_Other (e.g., "the Venus of Milo", "Genji Monogatari Emaki",...). Consequently, the performance of the FG-NER models for each category will be different. In this section, we investigate the performance some typical entity categories in the method **LSTM+CNN+CRF+Dict** (the best method for English).

Table 3c shows the Precision, Recall and F-score of the **LSTM+CNN+CRF+Dict** method 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 a small number of known entities (such as Country) 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 NE tag very well because these names might appear in various contexts. This result is consistent with the difficulty of the tasks as described in Section 2.2 and in Table 2.

4.4 Training data size and performance

To find out the relation between the training data size with the FG-NER models, we varied the training data size and measured the F-score of each model on the test dataset.

Specifically, we randomly sample the training dataset by the rates of 20%, 40%, 60%, 80% and 100% (i.e., using the entire training dataset). We then use the sampled training dataset to train the FG-NER models and evaluated these models with the test dataset. The results of this experiment are shown in



(a) For English FG-NER models

(b) For Japanese FG-NER models

Figure 4: Relation between training data size and F-score

Figure 4a and Figure 4b.

For English, we observe that the neural network-based method (**LSTM+CNN+CRF+Dict**) consistently outperforms the method **CRF+SVM+Dict** by a wide margin, irrespective of the training data size. If we only use 20% of the training data (i.e., about 4000 training sentences), we can get the F-score of 68.49%, whereas, using the entire of the training dataset (100%), we get the F-score of 83.14%. However, from 60% of the data size, the F-score increases gradually. This indicates that, we can still get a comparable performance even if we do not have a large amount of training data (an F-score of 78.68% for 60% of training data size).

For Japanese, the situation is totally different. When the training data size is small, **CRF+SVM+Dict** outperforms **LSTM+CRF+Dict+Cate** by a wide margin (F-score of 60.60%, compared to 45.43%). However, when the training data size is increased to 100%, **LSTM+CRF+Dict+Cate** is the best method (it achieves an F-score of 75.18%, compared to 73.30% of **CRF+SVM+Dict**). Therefore, if we only have a small amount of training data (e.g., 4,000 sentences), we must use **CRF+SVM+Dict**. If we have a larger amount of training data (e.g., 20,000 sentences), we can use the neural network-based model **LSTM+CRF+Dict+Cate** to achieve the best performance. When the amount of training data is large enough, the LSTM layer can learn the relation between the NE tags and the underlying words (tokens) as well as the grammatical structures. Consequently, it can precisely model the NE tagging problem.

These comparison results give us a suggestion to choose the appropriate model for each setting of target language and training data size: if the target language is English then we should use the neural network-based methods; if the language is Japanese then we should use **CRF+SVM+Dict** if we only have little amount of training data and we use the neural network-based method when we have enough amount of training data.

5 Conclusion

We presented an empirical study of fine-grained named entity recognition (FG-NER) methods. We investigated the relation between the performance of the methods with various settings of target languages and training data sizes. We found that the state-of-the-art method for English NER, which is based on neural network architecture, also works well with English FG-NER. However, for Japanese FG-NER, it does not achieve state-of-the-art performance. We proposed two additional features to the existing method: first, incorporating dictionary information to the model and second, utilizing entity category embeddings. Moreover, we proposed a modification to the architecture of the neural network-based model, that is removing the CNN layer for Japanese FG-NER to alleviate problem concerning a large number of characters in the Japanese alphabet. Experimental results show that, the proposed additional features and network architecture modification improve the performance of Japanese FG-NER by a wide margin.

References

Kurt D. Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008*, pages 1247–1250.

Nancy Chinchor. 1992. The Statistical Significance of the MUC-4 Results. In *Proceedings of the 4th Conference on Message Understanding, MUC 1992*, pages 30–50.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics (TACL)*, 4:357–370.

Cicero dos Santos and Victor Guimarães. 2015. Boosting named entity recognition with neural character embeddings. In *Proceedings of the Fifth Named Entity Workshop, NEWS 2015*, pages 25–33.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-Dependent Fine-Grained Entity Type Tagging. *CoRR*, abs/1412.1820.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016*, pages 260–270.

Xiao Ling and Daniel S. Weld. 2012. Fine-Grained Entity Recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2012*, pages 94–100.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, pages 1064–1074.

Andrew McCallum and Wei Li. 2003. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003*, pages 188–191.

Shotaro Misawa, Motoki Taniguchi, Yasuhide Miura, and Tomoko Ohkuma. 2017. Character-based Bidirectional LSTM-CRF with Words and Characters for Japanese Named Entity Recognition. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP, SCLeM 2017*, pages 97–102.

Nils Reimers and Iryna Gurevych. 2017. Reporting Score Distributions Makes a Difference: Performance Study of LSTM-networks for Sequence Tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017*, pages 338–348.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named Entity Recognition in Tweets: An Experimental Study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, EMNLP 2011*, pages 1524–1534.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning, CoNLL 2003*, pages 142–147.

Satoshi Sekine and Chikashi Nobata. 2004. Definition, Dictionaries and Tagger for Extended Named Entity Hierarchy. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 1977–1980.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended Named Entity Hierarchy. In *Proceedings of the Third International Conference on Language Resources and Evaluation, LREC 2002*, pages 1818–1824.

Satoshi Sekine. 2008. Extended Named Entity Ontology with Attribute Information. In *Proceedings of the Sixth International Conference on Language Resources and Evaluation, LREC 2002*, pages 52–57.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A Core of Semantic Knowledge. In *Proceedings of the 16th International Conference on World Wide Web, WWW 2007*, pages 697–706.

Masatoshi Suzuki, Koji Matsuda, Satoshi Sekine, Naoaki Okazaki, and Kentaro Inui. 2016. Fine-Grained Named Entity Classification with Wikipedia Article Vectors. In *Proceedings of the 2016 IEEE/WIC/ACM International Conference 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 Japanese). In *Proceedings of the Sixth Conference on Natural Language Learning, CoNLL 2002*, pages 119–125.

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.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding Methods for Fine Grained Entity Type Classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics, ACL 2015*, pages 291–296.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. HYENA: Hierarchical Type Classification for Entity Names. In *Proceedings of the 24th International Conference on Computational Linguistics, COLING 2012*, pages 1361–1370.

Guodong Zhou and Jian Su. 2002. Named Entity Recognition Using an HMM-Based Chunk Tagger. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, ACL 2002*, pages 473–480.