Neural Joint Learning for Classifying Wikipedia Articles into Fine-Grained Named Entity Types

Masatoshi Suzuki[†], **Koji Matsuda**[†], **Satoshi Sekine**[§], **Naoaki Okazaki**[†], **Kentaro Inui**[†]
† Tohoku University §Language Craft Inc
{m.suzuki,matsuda,okazaki,inui}@ecei.tohoku.ac.jp sekine@languagecarft.com

Abstract

This paper addresses the task of assigning fine-grained NE type labels to Wikipedia articles. To address the data sparseness problem, which is salient particularly in fine-grained type classification, we introduce a multi-task learning framework where type classifiers are all jointly learned by a neural network with a hidden layer. In addition, we also propose to learn article vectors (i.e. entity embeddings) from Wikipedia's hypertext structure using a Skip-gram model and incorporate them into the input feature set. To conduct large-scale practical experiments, we created a new dataset containing over 22,000 manually labeled instances. The dataset is available. The results of our experiments show that both ideas gained their own statistically significant improvement separately in classification accuracy.

1 Introduction

Recognizing named entities (NEs) in text is a crucial component task of a broad range of NLP applications including information extraction and question answering. Early work on named entity recognition (NER) defined a small number of coarse-grained entity types such as `Person` and `Location` and explored computational models for automatizing the task. One recent direction of extending this research field is to consider a larger set of fine-grained entity types (Lee et al., 2006; Sekine et al., 2002; Yosef et al., 2012; Corro et al., 2015). Recent studies report that fine-grained NER makes improvements to such applications as entity linking (Ling et al., 2015) and question answering (Mann, 2002). Given this background, this paper addresses the issue of creating a large gazetteer of NEs with fine-grained entity type information, motivated by the previous observations that a large-coverage gazetteer is a valuable resource for NER (Kazama and Torisawa, 2008; Carlson et al., 2009). Specifically, we consider building such a gazetteer by automatically classifying the articles of Wikipedia, one of the largest collection of NEs, into a predefined set of fine-grained named entity types.

The task of classifying Wikipedia articles into a predefined set of semantic classes has already been addressed by many researchers (Chang et al., 2009; Dakka and Cucerzan, 2008; Higashinaka et al., 2012; Tardif et al., 2009; Toral and Muñoz, 2006; Watanabe et al., 2007). However, most of these studies assume a coarse-grained NE type set (3 to 15 types). Fine-grained classification is naturally expected to be more difficult than coarse-grained classification. One big challenge is how to alleviate the problem of data sparseness when applying supervised machine learning approaches. For example, articles such as "Japan", "Mt. Fuji", and "Tokyo dome", may be classified as `Country`, `Mountain`, and `Sports_Facility` respectively in a fine-grained type set whereas all of them fall into the same type `Location` in a common coarse-grained type set. Given the same number of labeled training instances, one may obtain far fewer instances for each fine-grained type. Another challenge is in that fine-grained entity types may not be *disjoint*; for example, "Banana" can be classified as `Flora` and `Food_Other` simultaneously.

To address these issues, in this paper, we propose

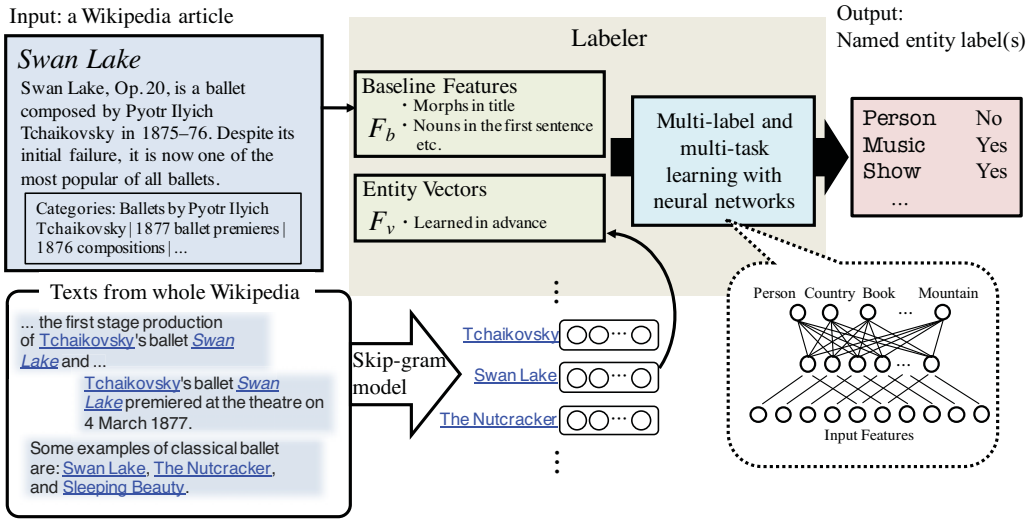


Figure 1: Automatic assignment of NE labels to Wikipedia articles based on multi-task learning and vector representation of articles

two methods (illustrated in Figure 1). First, we adopt the notion of multi-task learning (Caruana, 1997) and solve the whole task using a two-layered neural network. Our model learns all types of training instances jointly, which enables the model to learn combinations of input features commonly effective for multiple NE types with the hidden layer. By sharing effective feature combinations across different NE types, the data scarcity in minority NE types can be alleviated. Furthermore, this model can also naturally realize multi-label classification.

Second, we extend the feature set by exploiting the hyper-text structure of Wikipedia. The idea of using hyperlinks for Wikipedia article classification was first reported by Dakka and Cucerzan (2008). In this work, they represented local context of anchor texts of hyperlinks in Wikipedia as bag-of-words features. However, since its feature space was too sparse, they reported that the new context features had no effect on improving classification performance. Our proposal is to refine the context features using a distributed representation. To do this, we give each article a vector learned from all context words around hyperlinks (i.e. anchor texts) in Wikipedia using the Skip-gram model (Mikolov et al., 2013b). In the Skip-gram model, vector representations of words are learned so that two words similar in contexts have vectors with high similarity. In our intuition, articles in the same NE types are likely to be mentioned in similar contexts. Therefore, we adopt this model for learning article vectors.

We test our ideas on Japanese Wikipedia articles using the 200-NE type set proposed by Sekine et al. (2002). The results of our experiments show that the proposed methods achieve a 4.94-point improvement in entity-based F1 score. Our methods are particularly effective in labeling infrequent NE types.

Main contributions of this paper are as follows:

- We propose to apply a neural network-based multi-task learning method to the fine-grained multi-label classification of Wikipedia articles.
- We also propose to encode the local context of hyperlinks as vectors using the Skip-gram model. We make the obtained vectors publicly available[1].
- We created a new dataset by manually annotating over 22,000 Japanese Wikipedia articles with fine-grained NE types. The dataset is available if one contacts the authors.
- We tested our models on our new dataset and empirically showed their positive impacts on the accuracy of classification.

2 Related Work

The task of assigning labels of NE types to Wikipedia articles has been addressed in the context of automatic construction of an NE gazetteer

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

from Wikipedia articles. Toral and Muñoz (2006) proposed a method to classify Wikipedia articles into three NE types (`Location`, `Organization`, `Person`) using words included in the body of the article. They used WordNet as an external knowledge base for collecting hypernym information. They also applied weighted voting heuristics to determine NE types of articles. Dakka and Cucerzan (2008) classified articles into four NE types (`PER`, `ORG`, `LOC`, `MISC`) defined in ACE (Doddington et al., 2004) using supervised machine learning algorithms based on SVMs and naive Bayes. They used the bag-of-words in the target article as well as context words from the anchor text linking to the target article. Watanabe et al. (2007) focused on the HTML tree/link structure in Wikipedia articles. They formalized an NE categorization problem as assigning of NE labels to anchor texts in Wikipedia. They constructed graph-based representations of articles and estimated assignments of NE labels over the graphs using conditional random fields. In addition to these studies, there have been efforts toward automatic categorization, such as (Tardif et al., 2009; Chang et al., 2009). However, most of these studies assume a relatively small set of coarse-grained NE types (up to only 15 types).

In recent years, several projects such as YAGO (Suchanek et al., 2007) and DBpedia (Auer et al., 2007) have been devoted to provide Wikipedia articles with ontology class labels by applying simple heuristic or hand-crafted rules. However, these approaches heavily rely on metadata (e.g., infobox templates and category labels) and suffer from insufficient coverage of rules due to the lack of metadata, as reported by Aprosio et al. (2013).

Another trend of research which may seem relevant to our work can be found in efforts for automatically annotating entity mentions in text with fine-grained NE type labels defined in an existing type hierarchy such as Freebase (Ling and Weld, 2012; Nakashole et al., 2013; Shimaoka et al., 2016). While these studies focus on the identification and classification of individual mentions, our work aims at the classification of Wikipedia articles. The two tasks are related and may well benefit from each other. However, they are not the same; techniques proposed for mention classification cannot directly apply to our task nor can be compared with our methods.

The work closest to our study is done by Higashinaka et al. (2012), who proposed a supervised machine learning model for classifying Wikipedia articles into the 200 fine-grained NE types defined by Sekine et al. (2002). They conducted experiments to determine effective features extracted from article titles, body text, category labels, and infobox templates in Wikipedia. They train a logistic regression-based binary classifier for each type individually and the overall model chooses a single NE type receiving the highest score from the classifiers, ignoring the possibility that a Wikipedia article may belong to multiple NE categories. In contrast, our model learns classifiers for different NE types jointly and also addresses the issue of multi-label classification.

3 Data Preparation

3.1 Sekine et al.'s Fine-grained NE Type Set

In this study, we use the *Extended Named Entity Hierarchy*[2] proposed by Sekine et al. (2002) as our fine-grained NE type set. This ontology consists of 200 types, structured in a three-layered hierarchy. In this type hierarchy, a Wikipedia article may fall into multiple categories. Consider the following example:

> **Article title**: Godzilla
> **Article body**: Godzilla is a giant monster originating from a series of tokusatsu films of the same name from Japan. ... (excerpted from the English corresponding page of the same title)

It is reasonable to assume that the entity of this article belong to both `Character` and `Movie`).

3.2 Manual Annotation

From Japanese Wikipedia as of Nov. 23, 2015, we first extracted 22,667 articles that are hyperlinked at least 100 times from other articles in Wikipedia. We then manually annotated each of the 22,667 articles with one or more NE type labels from Sekine et al's type set[3].

Articles on abstract notions such as "Peace" and "Sleep" do not fall into any NE category particularly.

[2] `https://sites.google.com/site/extendednamedentityhierarchy/`

[3] The annotation was done by one annotator, supervised by the curator of Sekine et al's type set. Verification of the annotation accuracy is left for future work.

Table 1: 10 most frequent labels within the annotated dataset

Label name	Frequency	Example
Person	4,041	Isaac Asimov, Hillary Clinton, J. K. Rowling
Broadcast_Program	2,395	Sesame Street, Star Wars, Glee (TV series)
Company	1701	Sony, IBM, Apple Inc., Rakuten
City	975	New York, Tokyo, Melbourne
Product_Other	964	Microsoft Windows, Apple II,
Date	916	1977, January 3,
Book	909	Gutenberg Bible, The Lord of the Rings
Game	625	Lacrosse, Soccer, Table tennis
Pro_Sports_Organization	484	New York Yankees, Japan national baseball team
Position_Vocation	462	Physiotherapist, Prosecutor, Professor

Table 2: Infrequent labels within the annotated dataset

Frequency	Number of labels	Examples
0	55	URL, Temperature, Paintings
1	8	Ship, Star, Time
2-5	16	Canal, Market, Bridge
6-10	23	Earthquake, Treaty, School_Age
11-20	23	Public_Institution, Religious_Festival, Nationality

Table 3: Distribution of the number of labels per article

Number of labels assigned	Number of articles
1	21,624
2	850
3	187
4	14
6	2

We labeled such articles as CONCEPT. Wikipedia also includes articles or pages specific to Wikipedia like "List of characters in The Lion King" and "Wikipedia: Index". Those pages need to be discarded as well. We therefore decided to label such pages as IGNORED. Among our 22,667 articles, 2,660 articles are labeled as CONCEPT and 611 as IGNORED. Overall, our task is to classify Wikipedia articles into the 202 categories (Sekine et al.'s 200 types and the two additional categories).

Table 1 lists 10 most frequent labels that appear in the annotated articles and Table 2 shows examples of infrequent labels. As shown in these tables, the distribution of NE types in our data set is highly skewed. This makes the data sparseness problem salient particularly for the long tail of infrequent NE types.

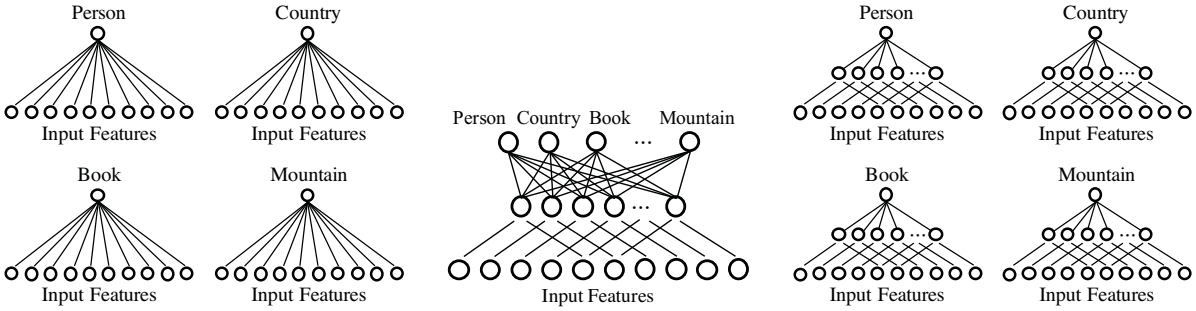
Table 3 shows the distribution of the number of labels assigned to one article in the annotated data. Most of the articles have only one label whereas 4.6% of articles were assigned multiple labels. This figure may seem not to be a big deal. However, given that the error rate of our model is already below 20% (see Section 5.2), considering the 4.6% is inevitable to seek further improvements.

4 Proposed Methods

4.1 Joint Learning of Multi-Label Classifiers

As a baseline approach of multi-label classification, we construct a classifier for each NE type; each classifier independently decides whether an article should be tagged with the corresponding NE type. We model this setting using binary classifiers based on logistic regression (Figure 2a). We call this model INDEP-LOGISTIC.

While INDEP-LOGISTIC is a simple model, this model may not work well for infrequent NE types because of the sparseness of the training data. This problem is crucial particularly in our task setting because the distribution of our NE types in Wikipedia is highly skewed as reported above. To address this problem, we propose a method based on multi-task learning (Caruana, 1997) and jointly train classifiers of all NE types. Concretely, we construct a neural network with a hidden layer (Figure 2b) and train it so that each node in the output layer yields the prob-



(a) INDEP-LOGISTIC (b) JOINT-NN (c) INDEP-NN

Figure 2: The three models for labeling types of articles.

ability of assigning the label of an NE type. Note that the activation function of the output layer is the singmoid function, not the softmax function. This means that this model can output mutiple labels of NE types for each article. In this method, we aim to learn effective combinations of input features which can also be used for labeling of infrequent NE types. We call this model JOINT-NN.

Note that there are two changes from INDEP-LOGISTIC to JOINT-NN: incorporating of a hidden layer and applying of joint learning. To examine the effect of each individual method separately, we also consider an intermediate model, INDEP-NN (Figure 2c). Similarly to INDEP-LOGISTIC, this model trains a classifier for each label, but has a hidden layer.

Formally, the INDEP-LOGISTIC model estimates the conditional probability that a given Wikipedia article represented by an n-dimensional feature vector $\boldsymbol{x} \in \mathbb{R}^n$ belongs to NE type c:

$$p_{\text{INDEP-LOGISTIC}}(y_c = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}_c \cdot \boldsymbol{x} + b_c), \quad (1)$$

where $\boldsymbol{w}_c \in \mathbb{R}^n$ and $b_c \in \mathbb{R}$ denote a weight vector and a bias term for NE type c, respectively. $\sigma(x) = \frac{1}{1+e^{-x}}$ is a sigmoid function.

The JOINT-NN model maps an input feature vector to a hidden layer with a matrix $\boldsymbol{W}$ whose parameters are shared across all the types:

$$p_{\text{JOINT-NN}}(y_c = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}_c \cdot \sigma(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) + b_c), \quad (2)$$

where $\boldsymbol{W} \in \mathbb{R}^{n \times k}$ and $\boldsymbol{b} \in \mathbb{R}^k$ denote a weight matrix and a bias vector of the k-dimensional hidden layer, $\boldsymbol{w}_c \in \mathbb{R}^k$ and $b_c \in \mathbb{R}$ denote a weight vector and a bias term, respectively, of the output layer, for each NE type c.

In contrast, the INDEP-NN model maps an input feature vector to a hidden layer by using a matrix W_c whose parameters are trained for each NE type independently:

$$p_{\text{INDEP-NN}}(y_c = 1|\boldsymbol{x}) = \sigma(\boldsymbol{w}_c \cdot \sigma(\boldsymbol{W}_c\boldsymbol{x} + \boldsymbol{b}_c) + b_c), \quad (3)$$

where $\boldsymbol{W}_c \in \mathbb{R}^{n \times k}$ and $\boldsymbol{b}_c \in \mathbb{R}^k$ denote a weight matrix and a bias vector, respectively, of the k-dimensional hidden layer for each NE type c. $\boldsymbol{w}_c \in \mathbb{R}^k$ and $b_c \in \mathbb{R}$ denote a weight vector and a bias term of the output layer for the NE type c, respectively.

The training data with N articles and C NE types is represented as $\{\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)}\}_{i=1}^N$, where $\boldsymbol{x}$ is a feature vector of an article and $\boldsymbol{y} = \{y_c\}_{c=1}^C$ is an array of binary variables indicating if the article belongs to an NE type c. With this data set, we minimize the cross entropy loss $\mathcal{L}$ of each model by using ADAM gradient-based optimization algorithm (Kingma and Ba, 2014):

$$\mathcal{L} = -\sum_{\boldsymbol{x},c} \{y_c \log(p(y_c = 1|\boldsymbol{x})) + (1 - y_c)\log(1 - p(y_c = 1|\boldsymbol{x}))\} \quad (4)$$

4.2 Input Features

We used two sets of features for building the models; one is a reproduction of the previous study (Higashinaka et al., 2012), and the other is our novel proposal.

4.2.1 Baseline Features

As a baseline feature set, we reproduced the features proposed by Higashinaka et al. (2012). Table 4 lists all of the basic features.[4] We were not

[4]Note that although the features of "Last n character(s) in the title" are effective in labeling NE types of Japanese article titles,

Table 4: List of features used for learning

Features
Word unigram of the title
Word bigram of the title
POS bigram of the title
Character bigram of the title
Last noun in the title
Last single character in the title
Last three characters in the title
Last character type in the title
Last noun in the first sentence
Headings of the article
Direct categories defined in Wikipedia
Upper categories defined in Wikipedia

able to reproduce features T8, T12, T14, and M22 described in the original paper (Higashinaka et al., 2012) because those features require the authors' internal resources to implement. For similar reasons, we used MeCab (Kudo et al., 2004) as a morphological analyzer instead of JTAG (Fuchi and Takagi, 1998), which was unavailable to us. For extracting text from Wikipedia dump, we used Wikipedia Extractor (`http://medialab.di.unipi.it/wiki/Wikipedia_Extractor`). We denote this baseline feature set as F_b.

4.2.2 Article Vectors

To extend the aforementioned basic feature set, we hypothesize that the way how each article (i.e. named entity) is mentioned in other articles can also be a useful clue for classifying that article. To test this hypothesis, we introduce distributed representations of Wikipedia articles.

Consider an article "Mount Everest". This article is hyperlinked from other articles as follows:

(1) *... After his ascent of Everest on 29 May 1953 ...*

(2) *... reached the summit of Everest for the twenty-first time ...*

(3) *... fatalities of the 2014 Mount Everest avalanche ...*

In this example, words near the anchor text, such as *summit* and *avalanche*, can be useful for estimating the semantic category of "Mount Everest" and assigning the label `Mountain` to the article "Mount Everest". While a number of approaches have been proposed for learning distributed representations of words, we simply adopt the Skip-gram model (Mikolov et al., 2013a) in this study.

Skip-gram trains a model so that it can predict context words from a centered word in a document. We apply this model to learn the embeddings of Wikipedia articles. To this end we need to address the following issues:

- An anchor text is not always identical to the article title to which the anchor refers. For this reason, we need to normalize an anchor text to the title of the article linked by the anchor.
- Article titles often consist of multiple words such as "White House". Therefore, we need a special treatment for tokenizing article titles.
- Not all of mentions to other articles are marked as anchor text in the Wikipedia articles. Typically, when an article mentions an entity multiple times, a hyperlink is inserted only at the first appearance of the mention to the entity[5].

To address these problems, we designed the following preprocessing steps. First, we replace every anchor text with the title of the article referred to by the hyperlink of the anchor text. Next, we assume all occurrences of the phrase identical to an anchor text to have hyperlinks to the article linked by the anchor text. This is based on the one-sense-per-discourse assumption. In addition, all white spaces in article titles are replaced with "_" to prevent article titles from being separated into words. In this way, we jointly learn vectors of words and articles. We use word2vec[6] to obtain 200 dimensional vectors.

We denote the 200-dimension article vector as F_v.

5 Experiments

To demonstrate the effectiveness of our models, we conducted experiments for labeling NE types to Japanese Wikipedia articles.

5.1 Settings

We tested the three classifier models (Indep-Logistic, Indep-NN, and Joint-NN) with two different feature sets (F_b and $F_b + F_v$). For each combi-

Higashinaka et al. (2012) reports that "Last two characters in the title" are not so useful in combinations with other features.

[5] `https://en.wikipedia.org/wiki/Wikipedia:Manual_of_Style/Linking`

[6] `https://code.google.com/p/word2vec/`

Table 5: Entity-based precision, recall and F1 of the models with different settings.

	F_b			$F_b + F_v$		
Model	Precision	Recall	F1	Precision	Recall	F1
INDEP-LOGISTIC	83.59	83.57	83.34	85.79	86.76	85.84
INDEP-NN	84.00	84.68	83.94	86.90	88.05	87.00
JOINT-NN (our model)	86.32	86.54	**86.14**	88.48	88.63	**88.28**

Table 6: NE labels whose weight vectors in output layers in (JOINT-NN, F_b) have high similarity to that of the NE label (in the header line of the table), accompanied with improvements of the label-based F1 score between (INDEP-NN, F_b) and (JOINT-NN, F_b). The number of articles assigned with an NE label is given in brackets.

	Label (# of articles)	$\Delta F1$	Label (# of articles)	$\Delta F1$	Label (# of articles)	$\Delta F1$
	`Book` (909)	5.28	`Country` (282)	1.38	`Food_Other` (57)	-0.13
Nearest Labels	`Broadcast_Program` (2395)	2.08	`Nationality` (14)	32.73	`Flora` (80)	5.04
	`Movie` (438)	3.64	`County` (126)	0.00	`Dish` (47)	10.36
	`Show` (43)	5.18	`Clothing` (12)	10.83	`Compound` (51)	8.12
	`Name_Other` (92)	0.53	`River` (58)	0.00	`Mineral` (12)	9.56
	`Printing_Other` (24)	4.64	`Island` (56)	3.72	`Religious_Festival` (12)	-5.92

nation of model and feature set, we evaluated classification performance by measuring entity-based/type-based precision, recall, and F1 value (Godbole and Sarawagi, 2004; Tsoumakas et al., 2009) over 10-fold cross validation. Entity-based precision, recall, and F1 value are calculated as below:

$$\text{Precision} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y^{(i)} \cap Z^{(i)}|}{|Z^{(i)}|} \quad (5)$$

$$\text{Recall} = \frac{1}{N}\sum_{i=1}^{N}\frac{|Y^{(i)} \cap Z^{(i)}|}{|Y^{(i)}|} \quad (6)$$

$$F_1 = \frac{1}{N}\sum_{i=1}^{N}\frac{2|Y^{(i)} \cap Z^{(i)}|}{|Z^{(i)}| + |Y^{(i)}|} \quad (7)$$

Here, $Y^{(i)}$ and $Z^{(i)}$ denote the set of correct labels and the set of predicted labels of article i, respectively. N denotes the number of documents. For type-based evaluation, we calculated precision, recall and F1 value of each named entity types.

For INDEP-LOGISTIC, we used scikit-learn (Pedregosa et al., 2011) to train classifiers. We used L2 penalty for regularization. For INDEP-NN and JOINT-NN, we used Chainer (Tokui et al., 2015) to implement neural networks. The dimension of the hidden layer was set to $k = 200$. When training the models, we used most frequent 10,000 baseline features (F_b) and 200-dimension article vectors (F_v) as input features of classifiers. For optimization, we used Adam with a learning rate of 0.001 and a mini-batch size of 10 and iterated over the training data until the cross-entropy loss per document gets smaller than 1.0×10^{-4}.

INDEP-LOGISTIC was implemented as a baseline model intended to reproduce the model proposed by Higashinaka et al. (2012). Note, however, that the results of our experiments cannot be compared directly with those reported in their paper because some of the features they used are not reproducible and the training/test data sets are not identical.

5.2 Results

The overall results are summarized in Table 5. We conducted binomial tests to determine statistical significance of the results, confirming that the improvement between any pair of settings is statistically significant $p < 0.01$ except that the improvement from (INDEP-LOGISTIC, F_b) to (INDEP-NN, F_b) was significant at $p < 0.05$.

Comparing the results between the baseline method (INDEP-LOGISTIC, F_b) and our full model (MULTI-NN, $F_b + F_v$), entity-based F1 score improved by about 5 points (83.34% to 88.28%), which is about 30% reduction of error rate. Table 5 also indicates that both of our two proposed methods, multi-task learning and article vector features, have

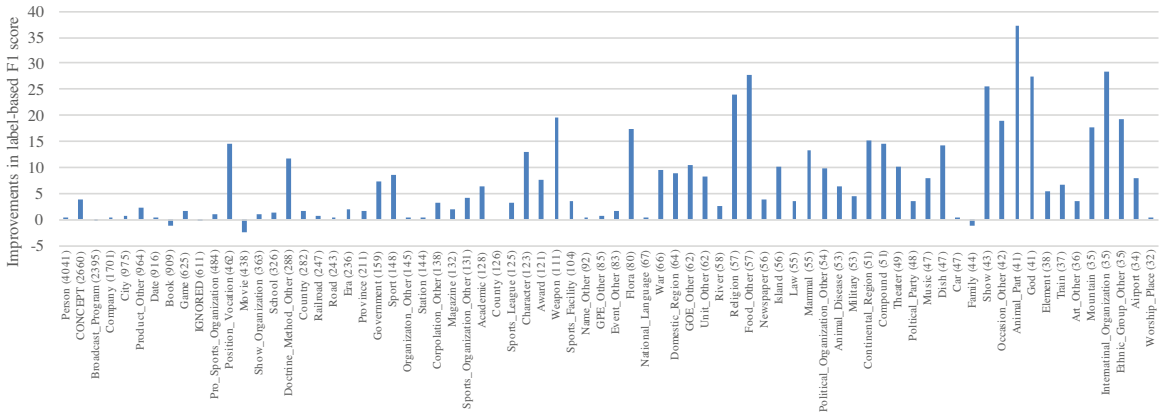


Figure 3: Improvement in F1 score per type between (INDEP-LOGISTIC, F_b) and (JOINT-NN, $F_b + F_v$). Only types with more than 30 (numbers are shown in brackets) articles are shown.

a separate significant gain.

To see the improvement in labeling performance per NE type, we compared label-based F1 score of each NE type between (INDEP-LOGISTIC, F_b) and (JOINT-NN, $F_b + F_v$). Figure 3 shows the improvement in F1 score for each NE type, where NE types are sorted by the number of articles in descending order. The figures indicate that our full model tends to obtain a larger gain particularly for infrequent NE types, which means our model addresses the data sparseness problem for infrequent NE types.

We made a deeper analysis of how our full model learns labeling of NE types. Our joint learning model is designed to learn combinations of features effective for multiple NE types. If two NE types share common combinations of features, they will have similar weight vectors at the output layer. So we observed clusters of the learned weight vectors at the output layer of (JOINT-NN, F_b) and discovered that many clusters comprise NE types that are semantically related with each other. Some example clusters are shown in Table 6. For example, the NE type `Book` has such neighbors as `Broadcast_Program`, `Movie` and `Show`. These NE types had similar weight vectors and gained considerable improvements with together. This demonstrates that our joint learning model learned combinations of input features and utilized them for multiple NE types effectively, which lead to the improvements observed particularly for infrequent NE types.

6 Conclusion

We have addressed the task of assigning fine-grained NE type labels to Wikipedia articles. To address the data sparseness problem, which is salient particularly in fine-grained type classification, we have introduced multi-task learning in which all the type classifiers are jointly learned by a neural network with a hidden layer. Additionally, to extend the input feature set, we have proposed to learn article vectors (i.e. entity embeddings) from Wikipedia's hypertext structure using the Skip-gram model and incorporate them into the input feature set. We created a new dataset containing over 22,000 manually labeled instances and conducted experiments on that dataset to evaluate the practical impacts of our ideas. The results show that both ideas gained their own statistically significant improvement separately in classification accuracy. The labeled dataset we created is available if one contacts the authors.

For future work, we aim to incorporate the hierarchy structure of NE types into classification. Also, each type in Sekine et al's NE type set has *attributes*. For example, `Mountain` has such attributes as *Height* and *People who reached the summit*. We aim to address a task of assigning correct attributes to each entity using the results of named entity classification.

Acknowledgments

This work was partially supported by *Research and Development on Real World Big Data Integration and Analysis*, MEXT and JSPS KAKENHI Grant 15H05318 and 15H01702.

References

Alessio Palmero Aprosio, Claudio Giuliano, and Alberto Lavelli. 2013. Extending the coverage of DBpedia properties using distant supervision over Wikipedia. In *Proceedings of ICON 2013*.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *Proceedings of ISWC'07/ASWC'07*.

Andrew Carlson, Scott Gaffney, and Flavian Vasile. 2009. Learning a named entity tagger from gazetteers with the partial perceptron. In *Proceedings of the 2009 AAAI Spring Symposium on Learning by Reading and Learning to Read*.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28:41–75.

Joseph Chang, Richard Tzong-Han Tsai, and Jason S. Chang. 2009. Wikisense: Supersense tagging of wikipedia named entities based wordnet. In *Proceedings of PACLIC 23*.

Luciano Del Corro, Abdalghani Abujabal, Rainer Gemulla, and Gerhard Weikum. 2015. FINET: Context-aware fine-grained named entity typing. In *Proceedings of EMNLP*, pages 868–878.

Wisam Dakka and Silviu Cucerzan. 2008. Augmenting wikipedia with named entity tags. In *Proceedings of 3rd IJCNLP*.

George Doddington, Alexis Mitchell, Mark Przybocki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. 2004. The automatic content extraction (ace) program tasks, data, and evaluation. In *Proceedings of LREC*.

Takeshi Fuchi and Shinichiro Takagi. 1998. Japanese morphological analyzer using word co-occurrence - jtag. In *Proceedings of ACL '98 and Proceedings of COLING '98*.

Shantanu Godbole and Sunita Sarawagi, 2004. *Advances in Knowledge Discovery and Data Mining: 8th Pacific-Asia Conference, PAKDD 2004, Sydney, Australia, May 26-28, 2004. Proceedings*, chapter Discriminative Methods for Multi-labeled Classification, pages 22–30. Springer Berlin Heidelberg.

Ryuichiro Higashinaka, Kugatsu Sadamitsu, Kuniko Saito, Toshiro Makino, and Yoshihiro Matsuo. 2012. Creating an extended named entity dictionary from wikipedia. In *Proceedings of COLING*.

Jun'ichi Kazama and Kentaro Torisawa. 2008. Inducing gazetteers for named entity recognition by large-scale clustering of dependency relations. In *Proceedings of ACL-08: HLT*, pages 407–415. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *ICLR*.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying conditional random fields to japanese morphological analysis. In *Proceedings of EMNLP*, pages 230–237.

Changki Lee, Yi-Gyu Hwang, Hyo-Jung Oh, Soojong Lim, Jeong Heo, Chung-Hee Lee, Hyeon-Jin Kim, Ji-Hyun Wang, and Myung-Gil Jang. 2006. Fine-grained named entity recognition using conditional random fields for question answering. In *Proceedings of Information Retrieval Technology, Third Asia Information Retrieval Symposium, AIRS 2006, Singapore, October 16-18, 2006*, pages 581–587.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *In Proc. of the 26th AAAI Conference on Artificial Intelligence*. Citeseer.

Xiao Ling, Sameer Singh, and Daniel S Weld. 2015. Design challenges for entity linking. *TACL*, pages 315–328.

Gideon S. Mann. 2002. Fine-grained proper noun ontologies for question answering. In *Proceedings of SEMANET '02*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient estimation of word representations in vector space. In *Proceedings of Workshop at International Conference on Learning Representations*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013b. Distributed representations of words and phrases and their compositionality. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Ndapandula Nakashole, Tomasz Tylenda, and Gerhard Weikum. 2013. Fine-grained semantic typing of emerging entities. In *51st Annual Meeting of the Association for Computational Linguistics*, pages 1488–1497. ACL.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of LREC*.

Sonse Shimaoka, Pontus Stenetorp, Kentaro Inui, and Sebastian Riedel. 2016. An attentive neural architecture for fine-grained entity type classification. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction (AKBC) 2016*.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of WWW*, WWW '07, pages 697–706, New York, NY, USA. ACM.

Sam Tardif, R. James Curran, and Tara Murphy. 2009. Improved text categorisation for wikipedia named entities. In *Proceedings of ALTA Workshop*, pages 104–108.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*.

Antonio Toral and Rafael Muñoz. 2006. A proposal to automatically build and maintain gazetteers for named entity recognition by using wikipedia. In *Proceedings of Workshop on New Text, EACL*.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2009. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer.

Yotaro Watanabe, Masayuki Asahara, and Yuji Matsumoto. 2007. A graph-based approach to named entity categorization in wikipedia using conditional random fields. In *Proceedings of EMNLP-CoNLL*.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard Weikum. 2012. HYENA: Hierarchical type classification for entity names. In *Proceedings of COLING 2012: Posters*, pages 1361–1370, Mumbai, India, December. The COLING 2012 Organizing Committee.