

## Evaluating and applying deep learning-based multilingual named entity recognition

Min-Ah Cheon<sup>1</sup> · Chang-Hyun Kim<sup>2</sup> · Ho-Min Park<sup>3</sup> · Jae-Hoon Kim<sup>†</sup>

(Received February 5, 2018 : Revised February 9, 2018 : Accepted February 13, 2018)

**Abstract:** Named entities (NEs) are words or phrases that are the referent of proper nouns with distinct meanings, such as a person (PER), location (LOC), or organization (ORG). Named entity recognition (NER) is a task that aims to locate and classify NEs in text into pre-defined categories such as PER, LOC, and ORG. NER is a well-studied area in natural language processing. Nevertheless, multilingual NER tasks that treat various languages in different language families with the same architecture are rarely investigated. In this paper, we discuss an NER system designed to deal with multiple languages. For our experiments, we develop Korean and Chinese NER systems. The experimental results show that the overall performance of the system in terms of the F-measure is 73.06% (Korean) and 40.67% (Chinese). Concurrently, the performance of NE detection has an accuracy of more than 94% for both Korean and Chinese. We apply the NER system conceptually for marine term extraction because the term extraction is similar to NER in that it detects words or phrases with specific meanings used in a particular context.

**Keywords:** Multilingual-named entity recognition, Marine term extraction, Deep learning

### 1. Introduction

According to Wikipedia [1], “A named entity is a real-world object, such as persons, locations, organizations, products, etc., that can be denoted with a proper name. ... Examples of named entities include *Barack Obama*, *New York City*, *Volkswagen Golf*, or anything else that can be named. Named entities can simply be viewed as entity instances (e.g., *New York City* is an instance of a city).” Named entities (NEs) are words or phrases that are the

referents of proper nouns with distinct meanings, such as a person (PER), location (LOC), or organization (ORG). Named entity recognition (NER) is a task that aims to locate and classify NEs in text into predefined categories such as PER, LOC, and ORG [2]-[4]. Various approaches to NER exist such as linguistic grammar-based techniques [3], statistics-based approaches [6]-[8], and deep-learning-based approaches [9][10]. Recently, the use of deep learning approaches has become prevalent in research, and it has a very high efficiency with regard to accuracy and time effort [11]. NER is a well-studied area in natural language processing. Nevertheless, a multilingual NER task is

rarely investigated; it treats various languages in different language families with the same architecture.

In this paper, we present an NER system that is designed to deal with multiple languages. A deep learning model of the NER system is designed for multiple languages, particularly Asian languages [12][13]. In this study, we evaluate the model in detail under multilingual environments. The system is then applied to marine term extraction because the process of recognizing NEs and the process of finding terms in documents are entirely identical. We expect that a multilingual NER system will be applicable for term extraction.

The remainder of this paper is organized as follows. Section 2 reviews related works on deep learning-based NER. Section 3 introduces a deep learning model for multilingual NER. In Section 4, we evaluate the performance of the proposed multilingual NER system based on some experiments. We discuss marine term extraction using the NER system in Section 5. In Section 6, the conclusions are drawn, and future research is discussed.

<sup>†</sup> Corresponding Author (ORCID: <http://orcid.org/0000-0001-8655-2591>): Department of Computer Engineering, Korea Maritime and Ocean University, 727 Taejong-ro, Yeongdo-gu, Busan 49112, Korea, E-mail: [jhoon@kmou.ac.kr](mailto:jhoon@kmou.ac.kr), Tel: 051-410-4574

1 Department of Computer Engineering, Korea Maritime and Ocean University, E-mail: [minah0218@kmou.ac.kr](mailto:minah0218@kmou.ac.kr), Tel: 051-410-4896

2 Language Processing Laboratory, Electronics and Telecommunications Research Institute, E-mail: [chkim@etri.re.kr](mailto:chkim@etri.re.kr), Tel: 042-860-6485

3 Department of Computer Engineering, Korea Maritime and Ocean University, E-mail: [homin2006@hanmail.net](mailto:homin2006@hanmail.net), Tel: 051-410-4896

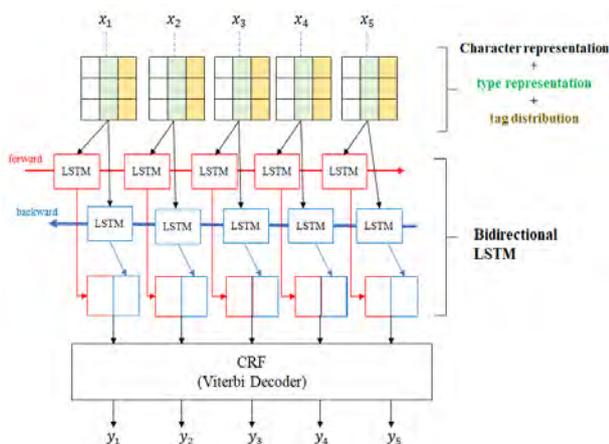
This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 2. Named Entity Recognition Overview

NER is a well-studied area in natural language processing, and generally, numerous results have been reported in the literature for most of the languages [2]-[4]. More recently, deep learning models have been shown to outperform other techniques applied for NER [10]-[14]. The effectiveness of the deep learning models is owing to their ability to extract and select features directly from the training dataset, instead of relying on handcrafted features developed from a specific dataset. Therefore, currently, deep learning approaches are widely used by researchers. Among deep learning models, bidirectional long short-term memory with a conditional random field layer (bi-LSTM/CRF) [9][12][13][15][16] has become highly sought after. The bi-LSTM/CRF approach combines the transition probability with the output of the bi-LSTM. This model has shown excellent performance for NER as well as any sequence labeling problem. Previous studies [9][16] have employed word embedding [12][13] or character embedding combinations [15]. These approaches use relatively more resources in addition to characters, (e.g., morphemes, part-of-speech, and the results of character LSTM), and they are difficult to apply in multilingual environments.

## 3. Deep Learning Model for Entity Recognition

Numerous deep learning models [9]-[13][15] for recognizing continuous entities such as NEs, terms, and chunks have recently been proposed and among these, the bi-LSTM/CRF model is very commonly used. We have also previously proposed a bi-LSTM/CRF model [12][13] for multilingual NER. In this article, we extend the model for Asian languages and term extraction. The overall architecture is as shown in **Figure 1**.



**Figure 1:** Architecture of entity recognition using bi-LSTM/CRF

### 3.1 Bi-LSTM/CRF Architecture for Entity Recognition

**Figure 1** shows the architecture of the bi-LSTM/CRF model. Bi-LSTM propagates input data bidirectionally to obtain output vectors that correspond to the probability of each output tag. The final results are calculated by adding the transition probability to the output vectors, which are the output tag sequences generated using bi-LSTM/CRF. The input is a vector that combines the look-up table values of character embedding, type embedding, and tag distribution, as described later in Sections 3.2–3.4. The input vector can be combined with dictionary information as shown in **Figure 1**, and the final output is the output tag as described subsequently in Section 3.5. The bi-LSTM layer can be stacked on multiple layers.

### 3.2 Character representation

A character used in the cultural area of Chinese characters (called hànzi, 漢字) represents a word or phrase. Collectively, these characters are known as CJK characters. Therefore, character embedding in a deep learning model for Asian languages can be more useful than that for European languages. Other advantages are that other tasks such as morphological analyses and POS tagging are unnecessary and that the robustness to unknown word problems is increased if the recognition unit is a character.

As shown in **Figure 1**, the input level of bi-LSTM/CRF for entity recognition is a character. Previous studies on character inputs [9][10] used character embedding that was of the same length as the longest word in a sentence. Our approach does not perform this combination; it instead combines all the information about the same character ( $x_n$ ). Character embedding is a collective name for a set of language modeling and feature learning techniques. Several methods are used for achieving character embedding [17]. This study uses the Word2vec models from the Python Gensim package [18].

### 3.3 Type representation

Asian countries such as Korea, China, and Japan have their own characters and writing systems while also using Chinese characters that have been partially transformed into their own forms by each country over time. In Unicode, Chinese characters including the transformed characters are collectively called CJK characters. These types of characters are useful for entity recognition. For example, consider a given sentence “국제해사기구(國際海事機構, International Maritime Organization, IMO)는 해운과 조선에 관한 국제적인 문제들을 다루기 위해 설립된 국제기구로, 유엔의 산하기관이다.” Typically, the NE appearing first in a text has much context represented by different character types. In the above example

sentence, “국제해사기구” is a NE, specifically an organization name, and is represented in Hangul, Hanja, and English.

Type embedding is almost identical to character embedding; the only difference is that the embedding unit is the first word of a Unicode type of the character. For example, the embedding unit of “국” is “HANGUL” because the Unicode type of “국” is “HANGUL SYLLABLE GUG” and the embedding unit of “국” is “CJK” because the Unicode type of “국” is “CJK UNIFIED IDEOGRAPH-570B”.

### 3.4 Tag distribution

Each character or character sequence has a different tag distribution. For example, a character sequence of “국제 (international)” has a high distribution of organization NE. In this study, we obtain the tag distribution from a training corpus that is publicly available or is crawled from the Web. **Equation (1)** defines the tag distribution, which is the probability that a specific character has a certain tag.

$$\text{tag distribution of } x_i = P(t_i|x_i) \tag{1}$$

where  $x_i$  is the  $i$ -th character of the input, and whether character or object  $x_i$  belongs to an NE or a term is determined with tag  $t_i$ .

### 3.5 Output tag

In character-based entity recognition, we face the usual problem of to the method for encoding NEs. Encoding involves character spans, such as tagging, which are sequences of tokens (words or other units) and tags (i.e., labels and categories). Consider the training corpus to be as follows:

“<ORG>국제해사기구</ORG>( <ORG>國際海事機構</ORG>, <ORG>International Maritime Organization</ORG>, <ORG>IMO</ORG>)는 해운과 조선에 관한 국제적인 문제들을 다루기 위해 설립된 국제기구로, 유엔의 산하기관이다.”

The NE “국제해사기구” consists of eight characters including spaces, and thus, each entity has a sequence of tags and not only one tag because the tag is for the characters that form the entity. Therefore, we need a scheme for encoding an entity with a sequence of character tags. We use the BIO tag representation for this purpose [19]. **Table 1** shows an example of entity encoding using the BIO tag representation.

**Table 1:** An Example of entity encoding using BIO tag representation

$x_i$	국	제		해		사		기	구	(
$y_i$	B-ORG	I-ORG	I-ORG	I-ORG		I-ORG	I-ORG	I-ORG	I-ORG	O

In **Table 1**,  $x_i$  and  $y_i$  are, respectively, an input character and an output tag. If a character does not belong to an entity, the character is assigned tag O. The B-prefix before an entity tag (e.g., ORG) indicates that the tag is at the beginning of an entity, and an I-prefix before a tag indicates that the tag is inside an entity. The B-tag is only used when a tag is followed by a tag of the same type without O tokens between them.

## 4. Performance Evaluation

### 4.1 Experimental data

We evaluated the performance of the multilingual NER model by performing an experiment with Korean and Chinese NER. **Table 2**

**Table 2:** Numerical information of the data used for Korean NER corpus and Chinese NER corpus

		Training Corpus	Validation Corpus	Test Corpus	Total
Korean NER corpus	# of NE sentences	16,621	2,077	2,077	20,775
	DAT (date)	7,083	844	854	8,781
	LOC (location)	9,889	1,260	1,206	12,355
	ORG (organization)	12,427	1,548	1,537	15,512
	OTH (other NE)	2,455	318	297	3,070
	PES (person)	12,787	1,611	1,625	16,023
	TIM (time)	762	90	113	965
Chinese NER corpus	# of NE sentences	1,348	268	268	1,884
	GPE.NAM	177	26	41	244
	GPE.NOM	0	1	0	1
	LOC.NAM	62	6	21	89
	LOC.NOM	28	6	8	42
	ORG.NAM	148	45	34	227
	ORG.NOM	17	5	11	33
	PER.NAM	522	90	96	708
PER.NOM	416	208	95	719	

presents the numerical information of the data used for each NER evaluation. The Korean NER corpus combines the ExoBrain NER corpus [20] provided by ExoBrain and our NER corpus built from news data. The Chinese NER used the data provided in [21].

#### 4.2. Evaluation measures

The performance evaluation unit is the chunk of each NE. The evaluation measures used in the experiments are precision, recall, and F1-measure. In **Equation (2)**, precision is given as the ratio of the correct NEs to the total NEs identified by the system. In **Equation (3)**, recall is given as the ratio of the total NEs identified by the system to the correct NEs. In **Equation (4)**, F-measure is the harmonic average of precision and recall.

$$precision = \frac{|system(NE) \cap correct(NE)|}{|system(NE)|} \quad (2)$$

$$recall = \frac{|system(NE) \cap correct(NE)|}{|correct(NE)|} \quad (3)$$

$$F - measure = \frac{2 \times precision \times recall}{precision + recall} \quad (4)$$

#### 4.3. Baseline systems

The baseline of Korean NER performance is the Korean NER using a CRF, which is a machine learning method. A CRF-based recognizer was developed as a baseline for bi-LST/CRF performance evaluation. The baseline of Chinese NER performance is the result of [22] using the same Chinese NER corpus.

#### 4.4. Independent variables

In this experiment, the independent variables are hyperparameters such as the dropout rate, batch size, learning rate, number of layers, and hidden nodes. In addition, we confirmed the input vector by combining the values given in Sections 3.2–3.4. The flag numbers according to the combination of input vectors are as follows.

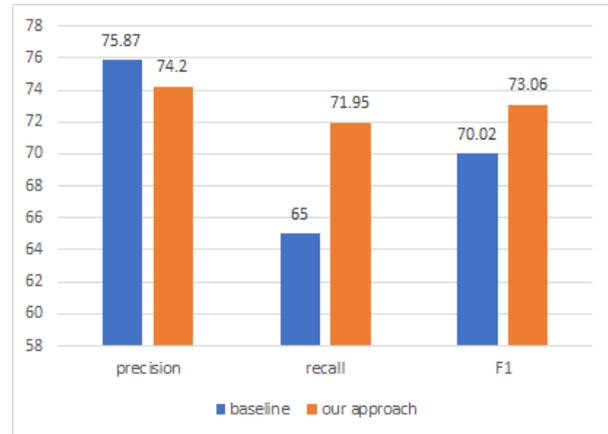
- (1) Character embedding
- (2) Character embedding + tag distribution
- (3) Character embedding + type embedding
- (4) Character embedding + tag distribution + type embedding

#### 4.5. Experimental results

##### 4.5.1. Korean NER

**Figure 2** compares the performance of the proposed method with the Korean NER baseline. It can be seen that the proposed method reduces the precision by 1.67% compared to the baseline,

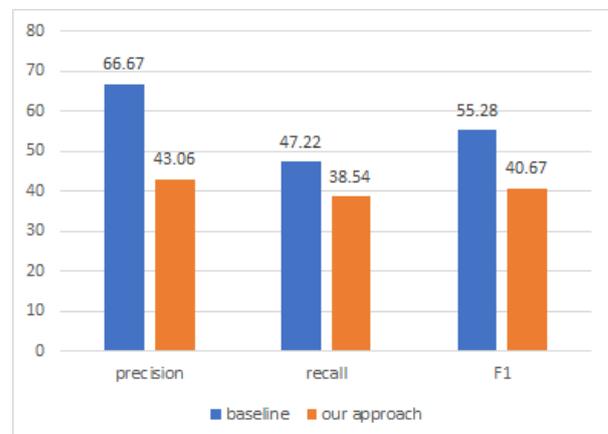
whereas it improves the recall by 6.95%. F1 also exhibits a 3.04% improvement over the baseline. We infer that a deep learning model is good at recall and a statistical model is good at precision. The input vector is flag 3, dropout rate is 0.7, batch size is 500, learning rate is 0.01, and there is a hidden layer and 256 hidden nodes.



**Figure 2:** Korean NER evaluation comparison baseline and our approach

##### 4.5.2. Chinese NER

Training data for Chinese NER are much smaller than that of Korean NER. This makes evaluation more difficult, but we are about to observe the performance. **Figure 3** compares the performance of the proposed method with the baseline for Chinese NER. The Chinese NER performance of the proposed method is relatively poor compared with the baseline. The precision is decreased by 23.61%, recall by 8.68%, and F1 by 14.61%. We observe that deep learning models require more training data compared with the statistical models. The input vector is flag 2, dropout rate is 0.7, batch size is 100, learning rate is 0.01, and there are two hidden layers and 256 hidden nodes in each hidden layer.



**Figure 3:** Chinese NER evaluation comparison baseline and our approach

4.5.3. Error analysis

The Chinese NER has very less training data and is very unstable, and therefore, an error analysis is not relevant. Therefore, we have not performed the error analysis for Chinese NER. In this section, we briefly mention the errors in Korean NER. **Table 3** presents the confusion matrix that summarizes the errors in the Korean NER. The column header is related to the results of the NER system, and the row header is related to the correct answers. The factors that most affect the precision adversely are the location and organization ambiguity. In the sentence “*The Vancouver Olympics took place in Canada in 2010,*” the NE category of “*Canada*” is the location. However, in the sentence “*Canada successfully attracted and held the Vancouver Olympics,*” “*Canada*” represents the organization. Thus, references to region and organization names are particularly difficult to distinguish. The next most noticeable error is that the system did not recognize the NE. It is assumed that this occurs because most output tags are deflected to the O tag. This causes a decline in the performance. We need to obtain an approach to solve the ambiguity of the NE category and the biased O tag. In both Korean and Chinese, the systems exhibit more than 94% accuracy for determining whether a word is an NE. Therefore, we expect that this model can be applied for term extraction and lexicon construction from text related to the domain of marine.

**5. Discussion and Application:  
Marine Term Extraction**

In this section, we discuss the applications of NER. Term extraction is similar to NER in that it detects words or phrases with specific meanings that are used in a particular context. Terminology is the study of terms and their use. Terms are words or phrases that have specific meanings in specific contexts or domains. Terms can be found using several natural language

processing and machine learning techniques [23]-[26]. An approach that extracts terms from a given corpus is called automatic term extraction. Approaches for automatic term extraction typically consist of two steps: candidate term generation and term selection. In the candidate term generation step, we identify noun phrases as term candidates using linguistic processors such as part of speech tagging and chunking. In the term selection step, we filter the candidate term list using statistical and machine learning methods to identify terms.

In this paper, we suggest marine term extraction as an application of NER, conceptually but not practically. Moreover, we discuss how to construct a dictionary of marine terms.

**Figure 4** shows the proposed term extraction system that automatically extracts relevant terms from a given corpus. We apply the system to the marine domain. In **Figure 4**, the dotted area is the part that uses a GUI tool and ① through ⑨ show the process of collecting text from the marine domain on websites, and then using the GUI tool to build a training corpus and marine terms semi-automatically. A semi-automatic NE tagger [27] can perform this task; the GUI tool needs to modify the annotation of the NE to classify terms. Raw corpora (A) and (B) are totally different for performance evaluation. The output of the html/xml tag remover is randomly merged among raw corpora (A) and (B). The solid line is a part of term recognition in a text by applying the NE recognizer and updating the term dictionary. The model in **Figure 4** is a result of the hyperparameter tuning for bi-LSTM/CRF using the training corpus. The trained model receives the data of raw corpus (B) as individual characters. The model refers to the pre-generated term dictionary and determines whether the input character is a term in a given domain such as marine. Then, the term dictionary is updated by the terms the model recognizes. Compared with the dotted line parts, the solid line parts have the advantage of being able to recognize multilingual

**Table 3:** Confusion matrix for errors in Korean NER

System Correct \	PES	LOC	ORG	DAT	TIME	OTH	O	TOTAL	Error rate (%)
PES		19	22				124	165	16
LOC	16		39				62	117	12
ORG	32	183				5	112	332	33
DAT					3	2	47	52	5
TIM							12	12	1
OTH	9	1	19		1		57	87	9
O	2	41	75	61	8	22		249	25
TOTAL	99	244	155	61	12	29	414	1,014	100
Error rate (%)	10	24	15	6	1	3	41		100

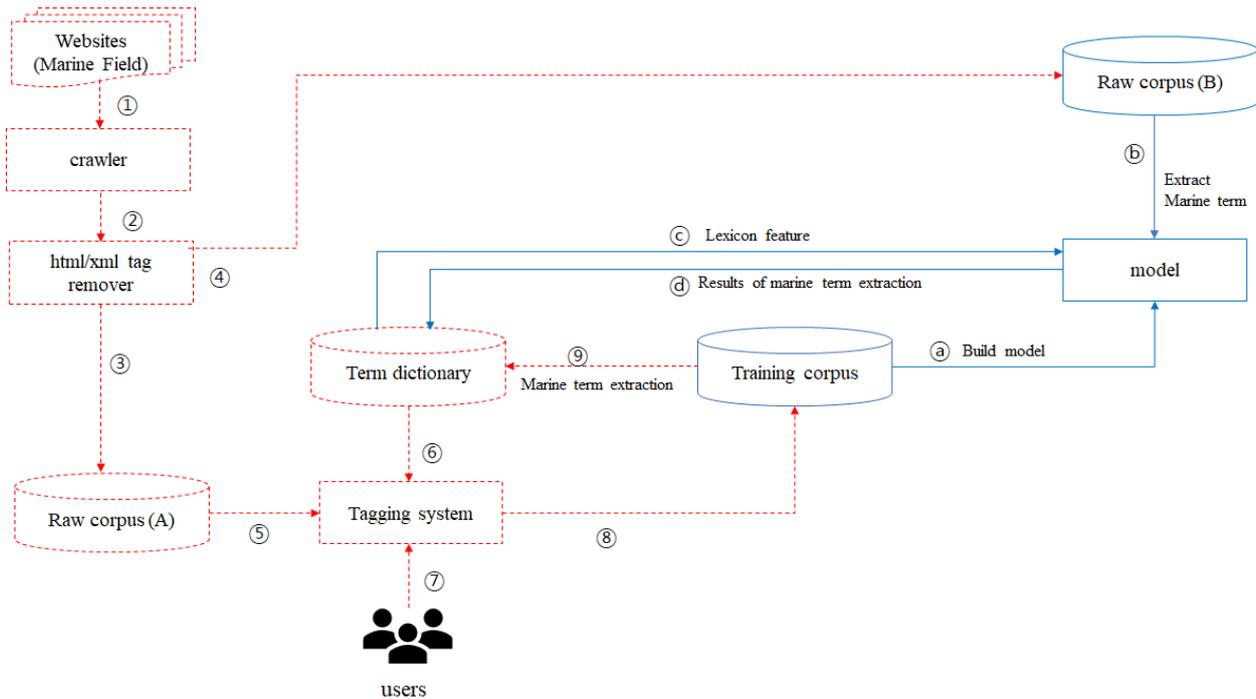


Figure 4: Overall architecture of term extraction system (when applied to the marine domain)

terms through training if sufficient multilingual training data exist. The key concept of this study is to recognize multilingual terms using the same architecture of bi-LSTM/CRF. Section 3 introduced the bi-LSTM/CRF architecture (model in Figure 1) for multilingual term extraction. Table 4 presents an example of term extraction using the bi-LSTM/CRF model in the marine domain. In Table 4, the tag encoding scheme is the same as that of the NER described in Section 3, i.e., the tags of B-M and I-M indicate the beginning and inside of a term, respectively. The tag of O indicates the outside of a term.

Table 4: An example of entity encoding using BIO tag representation

$x_i$	항	해	사	와		기	관	사	이	
$y_i$	B-M	I-M	I-M	O	O	B-M	I-M	I-M	O	O

A term dictionary can be constructed by repeatedly applying term extraction to newly added domain-related text. The size of the term dictionary gradually increases in proportion to the number of iterations.

### 6. Conclusion

In this work, we proposed a multilingual NER model (specifically for Asian languages) and evaluated its performance. For our experiments, we developed Korean and Chinese NER systems. Experimental results showed that the overall performance of the system in terms of F-measure was 73.06% (Korean) and 40.67% (Chinese). Concurrently, the performance

of NE detection was more than 94% accurate for both Korean and Chinese. We applied the NER to marine term extraction in a conceptual manner, because term extraction is similar to NER in that it detects words or phrases with specific meanings that are used in particular contexts. If multilingual NER is possible, multilingual term extraction will be possible through the same approach and would facilitate automatic dictionary building. Moreover, this approach will save effort and time costs for multilingual lexicon construction.

However, the limitations of this methodology are obvious. Our model supports multilingual recognition; however, it takes a long period of time to obtain the best hyperparameters of each language. In future research, we plan to study methods to automatically learn hyperparameters or share weights in multiple languages, in order to reduce such additional efforts. In addition, we will verify our hypothesis by applying a multilingual NER model to multilingual term extraction.

### Acknowledgements

This work was partly supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (R7119-16-1001, Core technology development of the real-time simultaneous speech translation based on knowledge enhancement) and the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2017M3C4A7068187)

## References

- [1] [https://en.wikipedia.org/wiki/Named\\_entity](https://en.wikipedia.org/wiki/Named_entity), Accessed January 20, 2018.
- [2] D. Nadeau and S. Sekine. "A survey of named entity recognition and classification," *Journal of Linguisticae Investigations*, vol. 30, no. 1, pp. 3-26, 2007.
- [3] K. Shaalan, "A survey of Arabic named entity recognition and classification," *Computational Linguistics*, vol. 40, no. 2, pp. 469-510, 2014.
- [4] D. Campos, S. Matos, and J. L. Oliveira "Biomedical named entity recognition: a survey of machine-learning tools," *Theory and Applications for Advanced Text Mining*, S. Sakurai (ed), InTech, 2012.
- [5] D. Farmakiotou, V. Karkaletsis, J. Koutsias, G. Sigletos, C. D. Spyropoulos, and P. Stamatopoulos, "Rule-based named entity recognition for Greek financial texts," *Proceedings of the Workshop on Computational Lexicography and Multimedia Dictionaries*, pp. 75-78, 2000.
- [6] D. M. Bikel, R. Schwartz, and R. M. Weischedel, "An algorithm that learns what's in a name," *Machine Learning*, vol. 3, no. 1-3, pp. 211-231, 1999.
- [7] A. E. Borthwick, *A Maximum Entropy Approach to Named Entity Recognition*, Ph.D. Thesis, New York University, NY, USA, 1999.
- [8] A. McCallum and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons," *Proceedings of the 7<sup>th</sup> Conference on Natural Language Learning*, vol. 4, pp. 188-191, 2003.
- [9] Z. Huang, W. Xu, and K. Yu "Bidirectional LSTM-CRF Models for Sequence Tagging," arXiv:1508.01991, 2015.
- [10] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dye, "Neural architectures for named entity recognition," arXiv preprint arXiv:1603.01360, 2016.
- [11] F. Démoncourt, J. Y. Lee, and P. Szolovits, "NeuroNER: An easy-to-use program for named-entity recognition based on neural networks," arXiv preprint arXiv:1705.05487, 2017.
- [12] M. A. Cheon, C. H. Kim, H. M. Park, K. M. Noh, and J. H. Kim, "Character-Aware Named Entity Recognition Using Bi\_LSTM/CRF," *Proceedings of the International Symposium on Marine Engineering and Technology*, pp. 126-128, 2017.
- [13] M. A. Cheon, C. H. Kim, H. M. Park, K. M. Noh, and J. H. Kim, "Multilingual named entity recognition with limited language resources," *Proceedings of the 29th Annual Conference on Human and Cognitive Language Technology*, pp. 143-146, 2017 (in Korean).
- [14] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa, "Natural language processing (almost) from scratch," *The Journal of Machine Learning Research*, vol. 12, pp. 2493-2537, 2011.
- [15] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bidirectional LSTM-CNNs-CRF," arXiv:1603.01354, 2016.
- [16] Y. H. Shin and S. G. Lee, "Bidirectional LSTM-CRNNs-CRF for Named Entity Recognition in Korean," *Proceedings of the 89th Annual Conference on Human & Cognitive Language Technology*, pp. 340-341, 2016 (in Korean).
- [17] S. Ruder, *Word embeddings in 2017: Trends and future directions*, <http://ruder.io/word-embeddings-2017/>, Accessed January 5, 2018.
- [18] <https://radimrehurek.com/gensim/index.html>, Accessed November 27, 2017.
- [19] L. A. Ramshaw and M. P. Marcus. "Text chunking using transformation-based learning," *Natural Language Processing Using Very Large Corpora*, S. Armstrong, K. Church, P. Isabelle, S. Manzi, E. Tzoukermann, and D. Yarowsky (eds), Springer, Dordrecht, pp. 157-176, 1999.
- [20] ExoBrain, [http://aiopen.etri.re.kr/service\\_api.php?loc=group02](http://aiopen.etri.re.kr/service_api.php?loc=group02), Accessed January 10, 2018.
- [21] <https://github.com/hltcoe/golden-horse/tree/master/data>, Accessed January 14, 2018.
- [22] N. Peng and M. Dredze, "Named entity recognition for Chinese social media with jointly trained embeddings," *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 548-554, 2015.
- [23] J. H. Oh, K. S. Lee, and K. S. Choi, "Automatic term recognition using domain similarity and statistical methods," *Journal of KIISE : Software and Applications*, vol. 29, no. 4, pp. 258-269, 2002 (in Korean).
- [24] J. H. Oh and K. S. Choi, "Machine-learning based biomedical term recognition," *Journal of KIISE : Software and Applications*, vol. 33, no. 8, pp. 718-729, 2006 (in Korean).
- [25] M. da S. Conrado, A. D. Felippo, T. A. Pardo, and S. O. Rezende, "A survey of automatic term extraction for Brazilian Portuguese," *Journal of the Brazilian Computer Society*, vol. 20, pp. 1-28, 2014.
- [26] A. Nazarenko and H. Zargayouna, "Evaluating term extraction," *Proceedings of the International Conference*

Recent Advances in Natural Language Processing, pp. 299-304, 2009.

- [27] K. M. Noh, C. H. Kim, M. A. Cheon, H. M. Park, H. Yoon, J. K. Kim, and J. H. Kim, "A semi-automatic annotation tool based on named entity dictionary," Proceedings of the 29th Annual Conference on Human and Cognitive Language Technology, pp. 309-313, 2017 (in Korean).