Pattern Recognition Letters
journal homepage: www.elsevier.com

Effective vector representation for the Korean named-entity recognition

ABSTRACT

Named-entity recognition, part of information extraction, is the task of finding the position of a proper names in a sentence and assigning it to the correct category. Existing studies have access to Korean named-entity recognition by a morphological-level method that performs named-entity recognition processes by using the results of morphological analysis as input. While this method has the advantage of using various linguistic clues, it suffers from the error propagation problem of morphological analysis. In this paper, we propose an effective methods for Korean syllable-level named-entity recognition to solve the above problem. Firstly, we suggest an approach to use the syllable bi-gram vector representation for Korean syllable-level named-entity recognition. Secondly, influenced by the linguistic characteristics of Korean, we suggest a novel way to make the joint vector representation of syllable bi-gram and Korean eojeol's positional information. In the experiment, we have evaluated our methods on the two Korean named-entity recognition corpora using Bi-directional LSTM-CRFs as a sequence labeler. Experimental results verify that our methods significantly improve the performance of syllable-level named-entity recognition and has similar performance to existing morphological-level named-entity recognition. Besides, additional experiments have shown that our syllable-level named-entity recognition is not only more robust but also faster than traditional morphological-level named-entity recognition by eliminating the morphological analysis process.

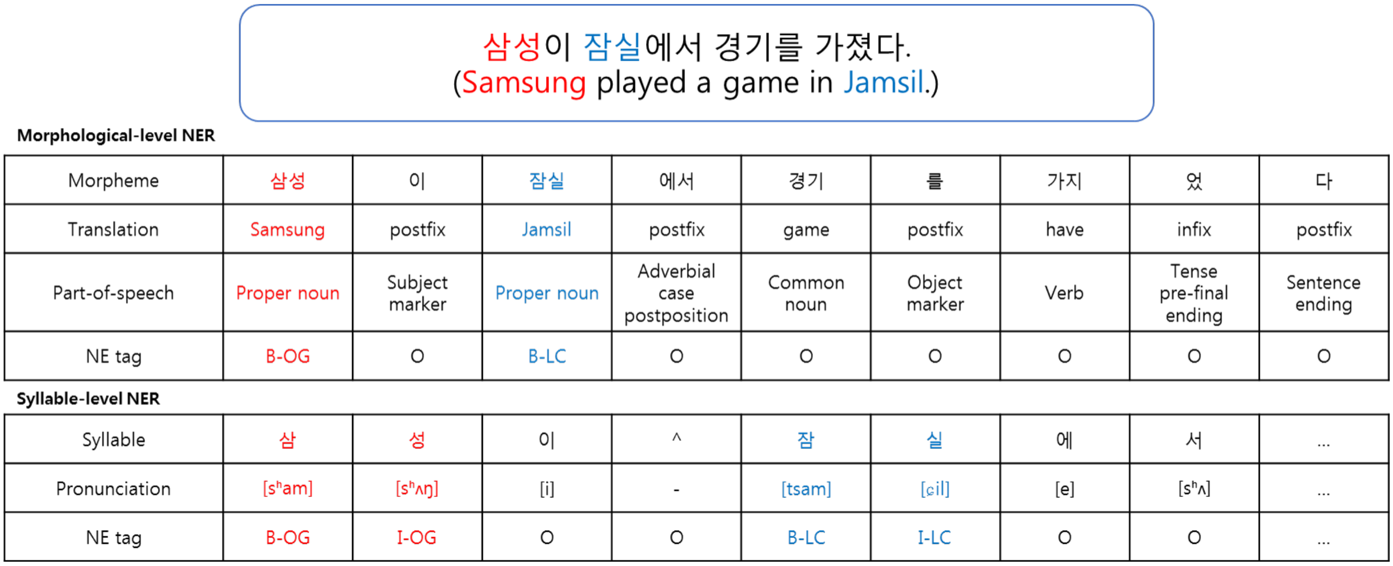
Keywords: Korean named-entity recognition; Syllable bigram vector representation; Deep neural network; Natural language processing

1. Introduction

Nowadays, with the development of the Internet, information extraction that analyzes useful information in text data becomes more and more important. Named-entity recognition (NER), which is a subfield of information extraction, is a technology for recognizing patterns of proper names in a document and categorizing them into an appropriate type. The term, Named-entity (NE), has been firstly coined for the 6th Message Understanding Conference (MUC-6) (Grishman and Sundheim, 1995) related to the information extraction (Nadeau and Sekine, 2007). Since then, NER was actively researched by leading research groups such as the CoNLL (Sang, 2002; Sang and Meulder, 2003) and ACE (Doddington et al., 2004). NE mainly contains proper names such as a names of person, location and organization, and numeric values such as time and date (Marrero et al., 2009). NER is now widely used in natural language processing (NLP), especially in fields where the proper name information has a great effect on the performance including dialogue system (Seon et al., 2012), knowledge base population (Dredze et al., 2010), to name but a few.

A corpus-based supervised learning method that trains with the NE labeled corpus is one of the most widely used method in NER studies. Early studies for the corpus-based supervised learning have trained NER sequence labelers using feature engineering and gazetteer (Zhou and Su, 2002; Saha et al., 2010). However, the cost to build different features and gazetteer at each domain is too expensive and it makes a NER system hard to be expanded into the other domain. Recently, to overcome this cost problem, it is becoming more and more popular to train NE patterns using word vector representation pre-trained from large unlabeled text (Collobert et al., 2011; Mikolov et al., 2013). In particular, Lample et al. (2016) reported that, with properly pre-trained word vector representation, it is possible to achieve even higher performance than existing methods with gazetteer features.

English can easily recognize boundaries between words through spacing, whereas Korean does not have explicit boundaries between words. Instead, an eojeol, the Korean spac-



삼성이 잠실에서 경기를 가졌다.
(Samsung played a game in Jamsil.)

Morphological-level NER

Morpheme	삼성	이	잠실	에서	경기	를	가지	었	다
Translation	Samsung	postfix	Jamsil	postfix	game	postfix	have	infix	postfix
Part-of-speech	Proper noun	Subject marker	Proper noun	Adverbial case postposition	Common noun	Object marker	Verb	Tense pre-final ending	Sentence ending
NE tag	B-OG	O	B-LC	O	O	O	O	O	O

Syllable-level NER

Syllable	삼	성	이	^	잠	실	에	서	...
Pronunciation	[sʰam]	[sʰʌŋ]	[i]	-	[tsam]	[ɕil]	[e]	[sʰʌ]	...
NE tag	B-OG	I-OG	O	O	B-LC	I-LC	O	O	...

Fig. 1. This figure demonstrates examples of Korean morphological-level NER and syllable-level NER. The meaning of each morpheme was translated into the second line, but we only describe the role of functional morphemes because it was difficult to be translated. The third line means the POS of each morpheme. Final line describes the NE tag of each morpheme as a IOB2 format (Sang and Veenstra, 1999). Meanwhile, first line of syllable-level NER is the sequence of syllables. In the next line, the pronunciation of each syllable is wrote as International Phonetic Alphabet (Coulmas, 2003). The final line stands for NE tags of each syllable. ‘OG’ represents the name of organization, ‘LC’ implies the place’s name, and ‘^’ symbolizes the border of eojeol.

ing unit, is combination of content and functional morphemes (Kang et al., 2003). For this reason, in most of Korean NER research, NER was accomplished by a pipeline-based approach attaching a NE tag to each morpheme based on the results of morphological segmentation and part-of-speech (POS) labeling as shown in Figure 1. This approach is known to have high performance since it can utilize abundant linguistic features. On the other hand, the morphological-level NER has frequently suffered from errors propagated from the morphological segmentation and the POS labeling process (Choi et al., 2012). Meanwhile, syllable-level NER attaches a NE tag to each syllable of the raw text as shown in Figure 1. This method free from the error propagation problem because it does not use morphological analysis.

This paper proposes a novel vector representation learning method for Korean syllable-level NER. The method consist of two parts. First of all, in order to solve the semantic ambiguity of syllable uni-gram, we suggest to utilize syllable bi-gram vector representation in the syllable-level NER. Secondly, we present a method to combine the eojeol’s positional information in the vector representation. In our experiments, our syllable-level NER with the proposed vector representation showed that its performance is significantly improved NER on Bi-directional LSTM-CRF (Bi-LSTM-CRF) (Lample et al., 2016). In addition, experimental results showed that our syllable-level NER method is comparable to the morphological-level NER in the environment where all morphological segmentation and POS labeling results are refined by human. Finally, our syllable-level NER method is faster than the traditional pipeline framework based the morphological-level NER method because of removing morphological analysis.

The paper is organized as follows. In chapter 2, we describe the related work. chapter 3 briefly introduces the concepts of Korean language appeared in this paper. We present our approaches in chapter 4. Chapter 5 is allocated to explain the neural network architecture of Bi-LSTM-CRF NER. Chapter 6 is devoted to the explanation of our experimental setting and results. Finally, the conclusions and future work are discussed in chapter 7.

2. Related work

Some previous studies to the Korean NER have tried to extract linguistic features sets suitable for a domain to train a sequence labeler (Lee et al., 2006; Choi et al., 2016). Lee et al. (2006) proposed linguistics features for the Korean NER and they trained CRFs with the feature set. Choi et al. (2016) suggested to utilize morpheme vector representations as features for CRFs and then they also clustered morpheme vector representations as an additional feature.

In recent years, Korean NER using deep learning has been actively studied (Yu and Ko, 2017; Nam et al., 2017). Yu and Ko (2017) suggested to extend the morpheme vector representation. They used not only traditional pre-trained morpheme vector representation but also additional information such as POS, gazetteer and the probability distribution of NE at each morpheme in the corpus. Finally, Nam et al. (2017) tried to preserve predicative particle information to the morpheme vector representation.

3. Characteristics of Korean language

This chapter briefly explains the characteristics of Korean required to understand this paper. In the first paragraph, we are going to characterize the Korean eojoel in terms of a sequence

Table 1. This is an example of separating an eojeol by proposal method. The example eojeol has same meaning of first eojeol of Fig 1.

Original eojeol	‘삼성이’
Separated as bi-grams	‘^삼’, ‘삼성’, ‘성이’
Attaching special tag	‘^삼’, ‘삼성_1’, ‘성이’

of syllables as a minimum speech unit. Second paragraph describes the Korean eojeol as a combination of morphemes that is the minimal unit of meaning.

Different from many other languages, Korean letters called *Jaso* do not compose text directly. Instead, two to five letters construct the phonological structure of syllable and consecutive syllables form eojeol. For example, Korean sentence in Figure 1 is made up of 4 eojeols ‘삼성이’, ‘잠실에서’, ‘경기를’, ‘가겼다.’. In this case, eojeol ‘삼성이’ can be divided as 3 syllables ‘삼’, ‘성’ and ‘이’, and a syllable ‘삼’ is compose of 3 letters ‘ㅅ’, ‘ㅏ’ and ‘ㅁ’.

Furthermore, in respects of linguistic typology, Korean is an agglutinative language forming sentence components by adding function morphemes to the end of a content morpheme (Lee and Ramsey, 2000). We can call the sequence of these morphemes as an eojeol or a morpheme combination (Kang et al., 2003). For instance, a proper noun ‘삼성’ and a subject marker ‘이’ organize the subject ‘삼성이’ of the example sentence in Figure 1.

4. Proposed vector representation for syllable-level NER

4.1. *Syllable bi-gram vector representation for Korean syllable-level NER*

Essentially, Korean syllables have a problem that their meaning is ambiguous. They may have a semantic meaning, represent a grammatical element, or a phonological element. For instance, Korean syllable ‘이’ can be euphony infix, a subject marker, as well as indicating teeth as noun. Some Korean NLP studies attempted to solve this problem by using syllable bi-gram statistics (Kang and Woo, 2001; Kwon et al., 2004). Inspired by these studies, we propose a method to improve the performance of Korean syllable-level NER using bi-gram vector representation.

The first eojeol ‘삼성이’ in the Table 1 can be divided into syllables [‘삼’, ‘성’, ‘이’]. In this case, the bi-gram can be constructed by combining uni-grams before each syllable, and the result is [‘^삼’, ‘삼성’, ‘성이’]. ‘^’ indicates the boundary of eojeol. If the sequence of NER labels for this syllable bi-grams are [‘B-OG’, ‘I-OG’, ‘O’], then ‘삼성’ is recognized as the name of organization. The vector representation of each syllable bi-gram is obtained by random initialization or pre-training and it is used input to the NER labeler. In this paper, the vector representation for syllable bi-gram is created by the word2vec skip-gram algorithm proposed by Mikolov et al. (2013).

4.2. *Joint vector representation of syllable bi-gram and eojeol's positional information*

In Section 4.1, we showed how to use syllable bi-gram vector representation for Korean NER to resolve ambiguity of Korean syllable. In reality, however, it is difficult to completely eliminate the semantic ambiguity problem of syllables by only using syllable bi-gram vector representation. For example, the Korean syllable bi-gram ‘라면’ means ramen noodle if it is noun, but it is also able to represent a postfix, meaning condition (‘if’).

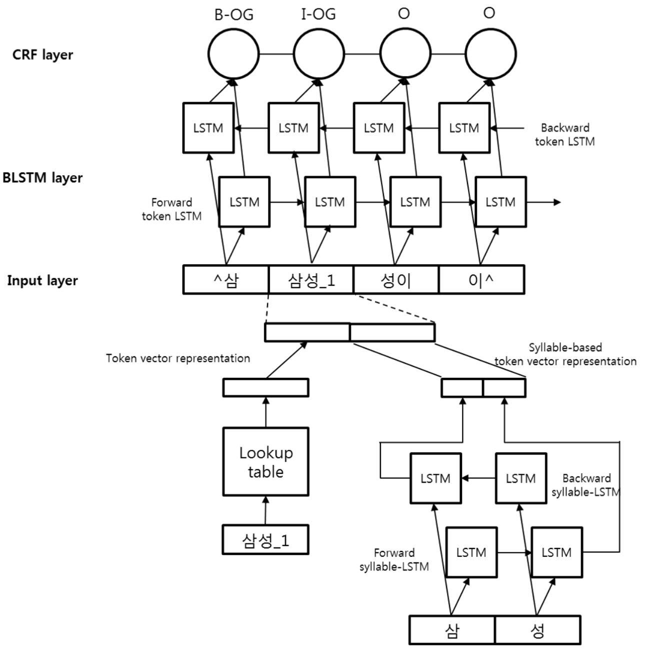


Fig. 2. Overall structure of Bi-LSTM-CRF NER labeler. This is an example of syllable bi-gram level NER and input tokens are syllable bi-grams.

The meaning of the syllable bi-gram can be disambiguated by its position and surrounding syllables. Thus, we pay attention to the morphological structure of Korean eojeol, which the content morpheme precedes the function morphemes as mentioned in Chapter 3. Since NE is a very typical content word, we make an assumption that the important clues for Korean NE pattern are in the front syllables of the eojeol. To confirm the above assumption, a special tag ‘_1’ is attached to distinguish the first syllable bi-gram of the eojeol from the other syllable bi-gram. Table 1. In this case, the same syllable bi-gram is handled as a completely different vocabulary according to the position of the each syllable bi-gram, and the syllable bi-gram vector representation is also created as completely different values.

5. Neural network structure for the NER labeler

As a NER sequence labeler, we choose Bi-LSTM-CRFs with state-of-the-art performance of NER in various languages including Korean (Yu and Ko, 2017; Lample et al., 2016). Figure 2 represents the structure of the Bi-LSTM-CRFs to predict the NE tag sequence $\mathbf{y}$ of length n for the input sentence X of length.

$$X = (x_1, x_2, ..., x_n) \tag{1}$$

$$\mathbf{y} = (y_1, y_2, ..., y_n) \tag{2}$$

Bi-LSTM-CRFs is made up of 3 different structures; Input layer, Bi-LSTM layer and CRF layer.

In the Input layer, each input tokens are converted to corresponding input vector representations. Input vector representation consist of two vector representations. First one, the token vector representation simply come from the lookup table that is initialized with random or pretrained vectors. Second one, syllable based token vector representation is the concatenation of final hidden of forward syllable-LSTM and backward syllable-LSTM. This is known to have a significant impact on the performance improvement in morphological-level NER. Lastly, the concatenation of token vector representation and syllable based token vector representation forms input vector representation.

Results of the input layer fed into the Bi-LSTM layer encoding contextual information $H = (h_1, h_2, ..., h_n)$. Here, in the Eq. 3, h_t means encoded vector of t-th input token. In addition, hf_t stands for t-th hidden layer of forward LSTM and hb_t stands for t-th hidden layer of backward LSTM. To encode left and right contextual information for each token, we concatenate hidden layer of forward LSTM and backward LSTM at each time step.

$$h_t = [hf_t; hb_t] \tag{3}$$

The NE tags are highly correlated between output tags. For example, the following tag for 'B-OG' as the begin of organization's name must be 'I-OG' or 'O'. Therefore, it is better to jointly decode the best sequence of NE tags rather than to decode each tag independently. In the CRF layer, we firstly calculate the observation score P of Eq.4. Then, by using the transition score matrix A, the score of the sequence y of the output NE tags for the input sentence X can be defined as Eq. 5. Finally, using the Viterbi search (Viterbi, 2010), we choose the highest score NE sequence $\hat{y}$ among all possible NE sequence Y_X's.

$$P = tanh(H \cdot W_o + b_o) \tag{4}$$

$$s(X, y) = \sum_{i=0}^{n} A_{y_i y_{i+1}} + \sum_{i=0}^{n} P_{iy_i} \tag{5}$$

$$\hat{y} = \underset{\tilde{y} \in \mathcal{Y}_X}{\operatorname{argmax}}\, s(X, \tilde{y}) \tag{6}$$

6. Experimental evaluation

In this chapter, we describe the experiments to demonstrate the effectiveness of the proposed method. Firstly, section 6.1 explains the experimental corpora and the evaluation measures. Section 6.2 presents the contents related to the experimental environment. In Section 6.3, we present various experimental results to verify the effectiveness of our proposed method.

6.1. Details of experimental data sets

In order to evaluate our NER method, we conducted in various experiments with two different NER corpora. First one is the Klpexpo2016 NER corpus released for the Korean language information processing system contest 2016[1]. Klpexpo2016 NER corpus is related to sport news domain and consist of 3,555 train sentences, 501 development sentences and 1,000 test sentences. The Klpexpo 2016 corpus has 12,372 manually tagged NEs of 5 categories. Second one is the ETRI language analysis corpus related to question-answering dialogue domain. The ETRI language analysis corpus is composed of overall 1,811 sentences and we conducted 10-fold cross validation. The ETRI language analysis corpus has 6,511 manually tagged NEs of 15 categories.

For the pretraining of vector representation, we randomly crawled about 2GB unlabeled corpus from the Korean news. The unlabeled corpus has approximately twelve million sentences and each sentence is made up of about 16 eojeols on average.

6.2. Experimental set up

Detailed structure and hyper parameter setting of Bi-LSTM-CRFs for the experiments is described in subsection 6.2.1. Additional experimental environments are shown in subsection 6.2.2.

6.2.1. Implementation details on NER labeler

The word2vec skip-gram algorithm was used to pretrain the vector representation of the lookup table with the following hyper-parameters. First, the vector dimensionality was set to 50 dimensions in the same way as the existing Korean NER studies (Kwon et al., 2016; Yu and Ko, 2017). The window size was set to 9, and all tokens that occurred less than 7 times in the corpus were treated as Out-of-vocabulary (OOV). The size of the negative sampling was set to 10, and the whole corpus was learned in 5 epochs. All the other hyper-parameters were set to default.

In the input layer of the Bi-LSTM-CRF NER labeler, the syllable-based token representation vector was set to 25 dimensions for forward and backward, respectively. This was added to the dimension 50 of the lookup table, and the input layer consisted of 100 dimensions. The hidden dimension of the Bi-LSTM layer was set to 100 dimensions along the dimension of the input representation. The model was trained by stochastic gradient descent (SGD) optimization algorithm with learning rate of 0.005 up to 100 epochs and the best model was selected with the development set. In addition, to prevent gradient explosion, the gradient was clipped (-5, 5) in the learning process. Finally, to avoid over fitting of the NER model, a dropout regularization (Srivastava et al., 2014) was applied between the input and Bi-LSTM layers at a drop rate of 0.5.

6.2.2. Other experimental environments

For the automatic morphological segmentation and POS labeling, we used the Komoran 2.4 morphological analyzer provided by Konlpy (Park and Cho, 2014). The experiments were conducted in Ubuntu 14.04 OS environment using Intel E5-2096 2.90GHz Xeon CPU and 128GB RAM.

As a performance evaluation measure of NER, we used a F_1-score criteria. The F_1-score of Eq. (9) is a harmonic mean of precision of a Eq. (7) and a recall of Eq. (8).

$$Precision = \frac{\#\, of\ true\ positive\ NEs}{\#\, of\ test\ outcome\ positive\ NEs} \tag{7}$$

[1] http://ithub.korean.go.kr/user/contest/contestIntroLastView.do

Table 2. Performance of the comparison test. The term "Random init." stands for lookup table is initialized to an arbitrary vectors otherwise "Pretrained init." stands for look up table is initialized to an pretrained vectors.

Model	Klpexpo 2016						ETRI language analysis					
	Random init.			Pretrained init.			Random init.			Pretrained init.		
	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1	Prec.	Rec.	F_1
Syll_1	74.09	73.19	73.64	73.75	71.18	72.44	63.01	59.70	61.31	62.83	60.11	61.44
Syll_2	79.51	76.83	78.15	83.41	80.62	81.99	73.99	67.45	70.57	76.58	71.26	73.82
Syll_3	79.74	75.65	77.64	84.39	82.40	83.38	74.02	67.86	70.81	77.69	71.94	74.71
Morph_1	81.42	79.63	80.51	85.17	82.43	83.78	74.34	71.09	72.68	76.61	73.33	74.93

$$Recall = \frac{\# of\ true\ positive\ NEs}{\# of\ true\ NEs} \tag{8}$$

$$F_1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

To determine the statistical significance of performance differences, we conducted a macro student t-test (Yang and Liu, 1999).

6.3. Experimental results

6.3.1. Effect of proposed syllable-level NER methods

To verify the effectiveness of our proposed method, we conducted experiments in the following four environments.

- **Syll_1**: This is a syllable-level NER baseline model in which the input layer consists of only syllable uni-gram embeddings from a lookup table.
- **Syll_2**: This is a syllable bi-gram NER model with bi-gram vector representation introduced in the section 4.1.
- **Syll_3**: This is a syllable bi-gram NER model with the joint vector representation of syllable bi-gram and the positional information of eojeol introduced in the section 4.2.
- **Morph_1**: This environment assumes ideal situations. For the training and test of NER model, the morphological analysis of input text was performed by human.

From the experimental results shown in Table 2, we found that the Syl_1, a syllable-level NER baseline, was poor in performance, and even initializing a pretrained syllable unigram vector representation does not help to improve performance. Meanwhile, Syl_2 significantly improved the syllable-level NER performance compared to Syl_1 ($p < 0.0001$) and its NER performance was improved when we initialized by the pretrained syllable bigram vector representation ($p < 0.001$). The results of Syl_3 are not significantly different from those of Syl_2 when arbitrarily initializing the input vector representation ($p > 0.2$). Otherwise, in the environment where vector representation is pretrained, the performance of Syl_3 is higher than that of Syl_2 in both of corpora ($p < 0.05$). In this result, we can find that the quality of the pretrained syllable bigram vector representation is improved when the positional information of eojeol is jointly utilized. Finally, the results of Syl_3 showed no significant difference compared to the results of morphological-level NER baseline, Morph_1 ($p > 0.1$).

Table 3. Experimental results of an environment considering error propagation in the Klpexpo2016 corpus.

Model	Random init.			Pretrained init.		
	Prec.	Rec.	F_1	Prec.	Rec.	F_1
Syll_1	79.74	75.65	77.64	84.39	82.40	83.38
Morph_1	81.42	79.63	80.51	85.17	82.43	83.78
Morph_2	71.20	71.86	71.53	72.71	74.25	73.47
Morph_3	77.62	73.47	75.49	80.00	79.72	79.86

6.3.2. Effect of error propagation in morphological-level NER in the Klpexpo2016 corpus in the Klpexpo2016 corpus

In the previous subsection, our syllable-level NER method (Syll_3) showed comparable performance to the morphological-level NER (Morph_1). At the test phase, Morph_1 analyzed the NER tags by using sentences that were manually annotated with the analyzed boundary and POS tag of each morpheme as input. However, in the actual environment, it is extremely rare that a input sentence is manually annotated. Instead, we mainly use the annotated results that are automatically generated by the morpheme analyzer as input. In this subsection, we attempt to evaluate the effect of morphological analysis error propagation on the morphological-level NER. For this, we conducted the following two experiments.

- **Morph_2**: The structure and training data of the model is the same as Morph_1. On the other hand, in the testing, NER is performed by input sentences analyzed automatically by the morphological analyzer in the testing.
- **Morph_3**: The structure of the model is the same as Morph_1. However, we use the input sentences that were automatically analyzed by the morphological analyzer for training and testing.

As shown in Table 3, when the morphological analysis errors are taken into account, the NER performance deteriorates sharply. As a result, our syllable-level NER method achieved about 3.2%p better than the morphological-level NER method in this real environment.

6.3.3. Comparison test in the Klpexpo2016 corpus

The proposed method were compared with other previous NER results with the same data set, Klpexpo2016: Choi's model(Choi et al., 2016), Yu's model (Yu and Ko, 2017) and Nam's model (Nam et al., 2017). Syll_3 was selected as our final model. In addition, gazetteer features that were used in

Table 4. F_1**-score Comparison of our model and existing morphological-level NER models in the Klpexpo2016 corpus.**

Model	Sequence labeler	Performance
Choi's model	CRFs	82.29
Yu's model	Bi-LSTM-CRFs	85.49
Nam's model	Bi-LSTM	84.73
Our model	**Bi-LSTM-CRFs**	**85.53**

Table 5. Results of executing time comparison test.

Model	Process	Time
Morphological-level NER	Morphological analysis	2,620 Sec.
	NER	1,558 Sec.
	Overall	4,179 Sec.
Syllable-level NER	NER	2,187 Sec.

the Choi's model and Yu's model were added to our model. Actually, this test is not a completely fair comparison. This is because the comparative models are the morphological-level NER models. Nonetheless, our result has similar or better performance to the performance reported by previous studies, as listed in Table 4.

6.3.4. *Comparison of execution time between traditional morphological NER model and our syllable-level NER model*

Since the NER system is often used in a large language analysis environment such as a search engine, the execution speed of the NER system is one of the important considerations. For this reason, we compared the executing time of our syllable-level NER with the morphological-level NER on a pipeline framework. We measured the running time of each NER systems, when using one million Korean sentences.

Table 5 demonstrates the comparison result of running time test. The experimental results show that our syllable-level NER model requires about 48% shorter executing time than the morphological-level NER model on a pipeline framework. Especially, morphological analysis itself is time consuming process requiring more time than our syllable-level NER. Note that, the executing time of syllable-level NER takes more time than the morphological-level NER because the syllable sequence of a sentence is longer than its morphological sequence.

7. Conclusions and future work

This paper has presented a novel approach to utilize the syllable information instead of the morpheme one for NER. The proposed approach achieved similar performance to morphological-level NER in the ideal environment that do not have the error propagation of morphological analysis and showed even much better performance than the morphological-level NER in the actual environment with error propagation from morphological analysis. And we have proposed a method to improve the Korean syllable-level NER performance with syllable bi-gram vector representations. In addition, inspired by a simple Korean language characteristic, we attempted to reflect positional information to the syllable vector representations.

Although our syllable-level NER is effective, there still is a room for improvement. Above all, when the boundary between the morphemes of an eojeol is ambiguous, our system often make an wrong segmentation. In order to solve this, the morphological structure of the eojeol need to be analyzed. We plan to investigate a multi-task learning method that a simultaneously train both morphological analysis and NER.

References

Choi, H., Kwon, S., Seo, J., 2016. Korean named entity recognition using clustered according to part of speech, in: Proceedings of HCI KOREA 2017, pp. 397–400.

Choi, J.Y., Kim, M.K., Park, S.Y., 2012. Named entity and event annotation tool for cultural heritage information corpus construction. Journal of the Korea Society of Computer and Information 17, 29–38.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P., 2011. Natural language processing (almost) from scratch. Journal of Machine Learning Research 12, 2493–2537.

Coulmas, F., 2003. Writing systems. An Introduction to Their Linguistic Analysis , 249–268.

Doddington, G.R., Mitchell, A., Przybocki, M.A., Ramshaw, L.A., Strassel, S., Weischedel, R.M., 2004. The automatic content extraction (ace) program-tasks, data, and evaluation., in: Proceedings of Language Resources and Evaluation Conference, pp. 837–840.

Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T., 2010. Entity disambiguation for knowledge base population, in: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 277–285.

Grishman, R., Sundheim, B., 1995. Design of the muc-6 evaluation, in: Proceedings of the 6th conference on Message understanding, pp. 1–11.

Kang, M.Y., Yoon, A., Kwon, H.C., 2003. Improving partial parsing based on error-pattern analysis for a korean grammar-checker. ACM Transactions on Asian Language Information Processing 2, 301–323.

Kang, S.S., Woo, C.W., 2001. Automatic segmentation of words using syllable bigram statistics., in: Natural Language Processing Pacific Rim Symposium, pp. 729–732.

Kwon, H.C., Kang, M.y., Choi, S.J., 2004. Stochastic korean word-spacing with smoothing using korean spelling checker. International Journal of Computer Processing of Oriental Languages 17, 239–252.

Kwon, S., Heo, Y., Lee, K., Lim, J., Choi, H., Seo, J., 2016. A korean named entity recognizer using weighted voting based ensemble technique, in: Proceedings of the 28th Annual Conference on Human & Cognitive Language Technology, pp. 333–336.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C., 2016. Neural architectures for named entity recognition, in: Proceedings of NAACL-HLT, pp. 260–270.

Lee, C., Hwang, Y.G., Oh, H.J., Lim, S., Heo, J., Lee, C.H., Kim, H.J., Wang, J.H., Jang, M.G., 2006. Fine-grained named entity recognition using conditional random fields for question answering, in: Asia Information Retrieval Symposium, pp. 581–587.

Lee, I., Ramsey, S.R., 2000. The Korean language. Suny Press.

Marrero, M., Sanchez-Cuadrado, S., Lara, J.M., Andreadakis, G., 2009. Evaluation of named entity extraction systems. Advances in Computational Linguistics, Research in Computing Science 41, 47–58.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013. Distributed representations of words and phrases and their compositionality, in: Proceedings of the Advances in neural information processing systems, pp. 3111–3119.

Nadeau, D., Sekine, S., 2007. A survey of named entity recognition and classification. Lingvisticae Investigationes 30, 3–26.

Nam, S., Hahm, Y., Choi, K.S., 2017. Application of word vector with korean specific feature to bi-lstm model for named entity recognition, pp. 147–150.

Park, E.L., Cho, S., 2014. Konlpy: Korean natural language processing in python, in: Proceedings of the 26th Annual Conference on Human & Cognitive Language Technology, pp. 133–136.

Saha, S.K., Narayan, S., Sarkar, S., Mitra, P., 2010. A composite kernel for named entity recognition. Pattern Recognition Letters 31, 1591–1597.

Sang, E., 2002. Introduction to the conll-2002 shared task: language-independent named entity recognition, in: Proceedings of the 6th conference on Natural language learning, pp. 1–4.

Sang, E., Meulder, F., 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition, in: Proceedings of the NAACL-HLT, pp. 142–147.

Sang, T., Veenstra, J., 1999. Representing text chunks, in: Proceedings of the ninth conference on European chapter of the Association for Computational Linguistics, pp. 173–179.

Seon, C.N., Kim, H., Seo, J., 2012. A statistical prediction model of speakers' intentions using multi-level features in a goal-oriented dialog system. Pattern Recognition Letters 33, 1397–1404.

Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. Journal of machine learning research 15, 1929–1958.

Viterbi, A.J., 2010. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm, in: The foundations Of the digital wireless world: Selected works of AJ Viterbi. World Scientific, pp. 41–50.

Yang, Y., Liu, X., 1999. A re-examination of text categorization methods, in: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 42–49.

Yu, H., Ko, Y., 2017. Expansion of word representation for named entity recognition based on bidirectional lstm crfs. Journal of KIISE 44, 306–313.

Zhou, G., Su, J., 2002. Named entity recognition using an hmm-based chunk tagger, in: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 473–480.