Korean-optimized Word Representations for Out-of-Vocabulary Problems caused by Misspelling using Sub-character Information

Seonhghyun Kim[1], Jai-Eun Kim[2], Seokhyun Hawang[3], Berlocher Ivan[4] and Seung-Won Yang[5]

[1-5] AI Labs, Saltlux Inc., Seoul, Republic of Korea
{seonghyunkim, jekim, shhwang, ivan, swyang}@saltlux.com

Abstract. In this paper, we propose Korean-optimized word representations that can better address the out-of-vocabulary (OOV) problem caused by misspelling. This problem is an important issue in many applications based on natural language processing. However, previous models do not fully consider the representations of misspelled OOV words. To overcome this problem, we propose sub-character information obtained from Korean Jamo units and also adopt additional sub-character information to better withstand the misspelling. Finally, experimental results show that our model is about 2.3 times more accurate than the conventional model in case of the misspelled word while still maintaining the semantic relationship of the words.

Keywords: Word embedding, Word representation, Machine learning Korean, Out-of-vocabulary, Misspelling, Sub character, Natural language processing.

1 Introduction

Continuous word representations play a major role in many natural language processing (NLP) applications based on neural networks approaches such as named entity recognition (NER) or machine reading comprehension (MRC). Previous studies that measure the semantic relatedness between words using word embedding, such as Word2Vec [1, 2], have been successfully implemented in these applications [3, 4].
However, previous word representations have some limitations. First, they allow computing word vectors for only words that appear in the training data. Second, the linguistic characteristics of languages are not considered for the word representation. In most languages, the semantics of a word are determined by combining subword information, such as morphemes, syntax, and root.

To overcome these problems, recent studies [5, 6] have considered the subword information represented as a bag of character n-grams. Thus, word representations can be more effectively learned by embedding subword information. Moreover, these models calculate the vector of the first word to be seen, often called the out-of-vocabulary (OOV), by using character n-grams. However, directly utilizing subword information from character n-grams would be vulnerable to misspelled words in some

languages based on a featural writing system, such as Korean [7]. This is because misspelling, even of a single character, is a critical factor to change the semantics of subword in a featural writing system [8].

In this paper, we focus on the Korean language. Korean is unique in that each character within a word is composed of two or three sub-character units called Jamo [9, 10]. In this work, we propose optimized word representations for Korean that utilize the proposed sub-character units. We verify that the proposed word representations are better to address the OOV problem than the existing public model for misspelled words while still maintaining a good semantic relationship between words.

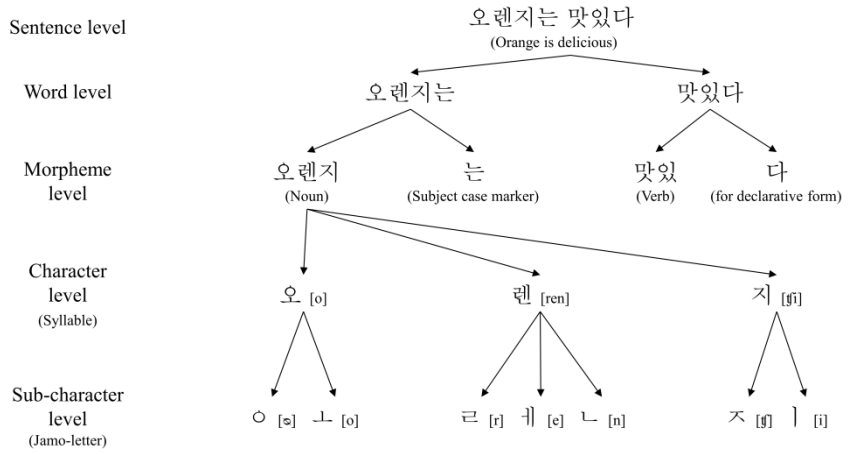

Fig. 1. Hierarchical structure of Korean.

2 Related Works

Many studies on word representations have used a morpheme as the subword information because it is the smallest unit of meaning in linguistics [1, 2, 11-13]. References [1, 13] proposed a continuous bag of words (CBOW) model that predicts the current word from the surrounding context words and a skip-gram model that uses the current word to predict the surrounding context words. Reference [2] proposed a GloVe vector model that utilizes the probability of the co-occurrence between words. However, these methods cannot solve the OOV problem. The Facebook AI Research group proposed a word representation method [6, 14, 15] called FastText, which utilizes subword information. This method solves the OOV problem by vectorizing with subwords. According to their results, OOV words also can be vectorized with good semantic relations because subword (character n-grams) vectors contain semantic characteristics. The cosine similarity between -adolesc- from preadolescent (OOV word) and young shows a high semantic relation. However, in our experiments, when OOV occurs by misspelling, word similarity is significantly lower than expected because a subword character is changed.

3 Methods

In this section, we describe an efficient word representation method for the Korean language directly inspired by previous research [6]. Their model can construct OOV word vectors using the vector summation of syllable-based n-gram subword units. In contrast to many other languages, Korean words are not just a concatenation of characters by syllable units [9]. Our main contribution is to propose optimized word representations for Korean by considering its featural characteristics.

3.1 Jamo-static Model

To illustrate the Korean linguistic features, we show the diverse linguistic levels for the sentence, 오렌지는 맛있다 (Orange is delicious) in Fig. 1. This sentence is constructed using two words 오렌지는 (Orange) and 맛있다 (is delicious). Words can be decomposed into the morpheme level and the character level.

In the case of 오렌지 (Orange), the word consists of three characters: 오, 렌, and 지. Each character consists of two or three sub-characters (Jamo), which are an initial consonant and a vowel or sometimes a consonant placed under a vowel. The sub-character level is the basic unit of Korean, and it is an important unit for word representation.

To learn word representation based on these Korean characteristics, we tokenize a word into a morpheme unit using special symbols, < and >, at the beginning and end of the words in the same manner as a previous subword model [6]. The morpheme units are decomposed to the character level. Each character is disassembled into sub-characters. We also add a special boundary symbol, ., between the characters as a mark to separate them. Taking the word, 오렌지 (Orange), as an example, it can be represented by sub-characters as follow:

<ㅇㅗ.ㄹㅔㄴ.ㅈㅣ.>

We first conducted an experiment based on this sub-character as a Jamo-static model.

3.2 Jamo-advanced Model for Misspelling

In Korean, most misspelling problems occur at the sub-character level, especially a vowel, due to similar pronunciations or typos. For example, 렌 [ren], ㅖ [e] is a vowel that can be written using a similar pronunciation, such as 랜 [ræn], 랸 [ryæn], or 롄 [ryen].

This type of misspelling causes OOV problems even though the corrected word is part of the vocabulary. Although the meaning of a word and the subword information are completely distorted by the misspelling, this problem is very common in Korean.

Therefore, a novel method is required to correct the semantic distortion and to preserve the subword information of misspelled OOV words.

In our method, we design the additional sub-character *n*-gram vectors that contain a set of sub-characters without a vowel for each character. As with the word 오렌지 (Orange), each vowel is excluded as ㅇ렌지, 오ㄹㄴ지, and 오렌ㅈ, and the additional sub-characters are:

Without ㅗ: <ㅇ ㅣ ㄹ ㅔ ㄴ ㅣ ㅈ ㅣ ㅣ>
Without ㅔ: <ㅇ ㅗ ㅣ ㄹ ㄴ ㅣ ㅈ ㅣ ㅣ>
Without ㅣ: <ㅇ ㅗ ㅣ ㄹ ㅔ ㄴ ㅣ ㅈ ㅣ>

In order to compare how this method affects word representation in OOV, we conducted an experiment with Jamo-static model that learns the word vector based on the sub-character unit. Next, we adapted the additional sub-character vectors as a Jamo-advanced model.

4 Experimental Setup

4.1 A Model Variations

Baseline. The pre-trained published word vector model based on character units [6].
Jamo-static. A model that learns the word vector based on the sub-character units.
Baseline. The Jamo-static model noted above but with our additional sub-character *n*-gram vectors.

4.2 Datasets

We trained our model on the same parameters and training data (Wikipedia dumps of September 2016) used in previous research [6]. In order to evaluate the word representations for misspelled words, we designed a new comprehensive test set containing 250 paired OOV words consisting of misspelled OOV words and corrected words.

4.3 Implementation Details

To tokenize the Wikipedia dumps into morpheme units, we used open API software for morpheme analysis in Korean[1]. Our model embeds words that appear at least five times in the training set into a 300 dimensional space. We use a 1-5 context window size and a 10-4 rejection threshold. The step size is set to 0.025 (for more details, see [6]). The *n*-gram range is set to 6-12 and includes 2-3 syllables with special boundary symbols.

[1] http://www.adams.ai/apiPage?tms=pos

5 Results

5.1 Correlation with Human Judgement

We first evaluated the semantic relationship between words by computing Pearson's correlation coefficient between human judgement and the cosine similarity of the vector representations. We used the translated WordSim353 dataset [16].

Table 1. The results of translated WordSim353.

Dataset	Model	Correlation
WordSim353	Baseline	0.726
WordSim353	Jamo-static	0.745
WordSim353	Jamo-advanced	0.744

As shown in Table 1, the semantic relation of our Jamo-static and Jamo-advanced models is as strong as the baseline model. We also noticed that our proposed model is more highly correlated with human judgement than the baseline model.

5.2 Similarity between Misspelled Words and Corrected Words

Cosine Similarity. Next, we compared the cosine similarity between the misspelled OOV words and the corrected words. If the misspelled word is semantically related to the corrected word, it will show a high cosine similarity. Statistical significance was tested using Student's t-test.

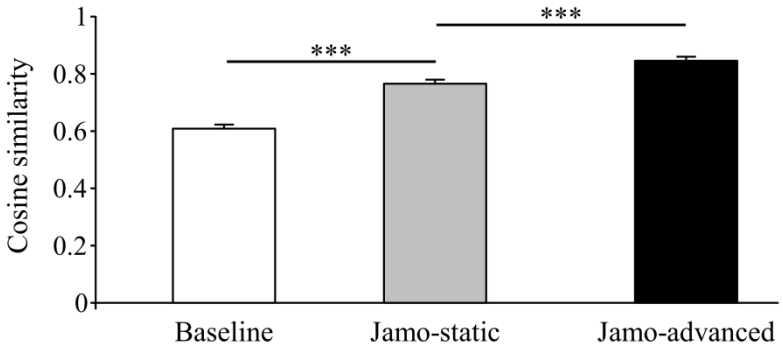

Fig. 2. Cosine similarity between a misspelled word and a corrected word.

Consequently, both the Jamo-static and Jamo-advanced models showed higher similarity scores than the baseline model (Fig. 2). Moreover, the difference between the Jamo-advanced model and the Jamo-static model was statistically significant (P <

0.001). These results indicate that subword information is better reflected in the word representation based on sub-character units than word units.

Most Similar Words. Finally, to evaluate how close the misspelled word is to the corrected word in the word vector space, we observed the 10 words that are most similar to the misspelled words. The accuracy of the proposed method was determined by checking whether the corrected word was included in the words that are most similar based on the range of words.

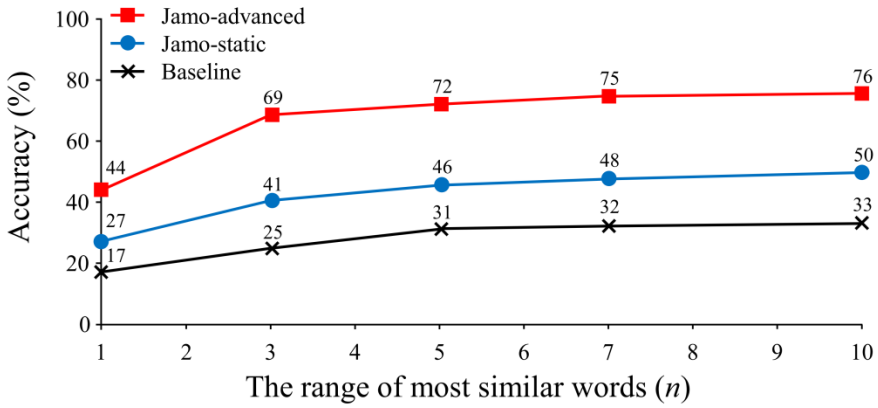

Fig. 3. The ratio that the corrected word is included in the top *n* most similar words from the misspelled OOV words.

As shown in Fig. 3, the baseline model was found to be less than 50% accurate over the entire range. The accuracy was greater for the Jamo-static model than the baseline model. When we observed the 10 words that are most similar, the corrected word was included in more than half of the test words. The accuracy was dramatically increased for the Jamo-advanced model. In contrast to the most similar word in an OOV word exactly matched its corrected word with 17% accuracy for the baseline model; Jamo-advanced model shows 44% accuracy. When the number of words in the most similar range was increased to 10, the accuracy increased to 76%, which is approximately 2.3 times higher accuracy than the baseline model with 33% accuracy.

Table 2 shows the top 3 most similar words of the baseline model and Jamo-advanced model as an example with misspelled OOV words (bold text). In the vocabulary word, the baseline model and the Jamo-advanced model show similar words in top 3, while in the OOV words, the baseline model shows quite unrelated most similar words compare to Jamo-advanced model. Moreover, the corrected words are contained in the most similar words of top3. These results indicate that the misspelled OOV word was successfully represented the semantic relationship of the words as the original corrected word by using our Jamo-advanced model.

Table 2. Top 3 neighboring words based on cosine similarity for some example words that contain misspelled OOV words (bold text) in baseline model and Jamo-advanced model.

Input word	Model	Most similar words in range of top 3		
페이스북 (Facebook)	Baseline	트위터 (Twitter)	공식페이스북 (Official Facebook)	인스타그램 (Instagram)
	Jamo-advanced	트위터 (Twitter)	SNS (Social Media)	인스타그램 (Instagram)
월트디즈니 (Walt Disney)	Baseline	디즈니 (Disney)	디즈니툰 (DisneyToon)	디즈니주니어 (Disney Junior)
	Jamo-advanced	디즈니툰 (DisneyToon)	디즈니채널 (Disney Channel)	디즈니사 (Walt Disney Company)
타자 (Hitter)	Baseline	톱타자 (Lead-off man)	좌타자 (Left-handed hitter)	다음타자 (Next hitter)
	Jamo-advanced	우타자 (Right-handed hitter)	좌타자 (Left-handed hitter)	장타자 (Power hitter)
페날티 **(Penalty)** **OOV word**	Baseline	리날디 (Rinaldi)	페레티 (Ferretti)	마셰티 (Machete)
	Jamo-advanced	페널티골 (Penalty goal)	**페널티** **(Penalty)**	드록바 (Drogba)
나푸탈렌 **(Naphthalene)** **OOV word**	Baseline	야렌 (Yaren)	콜루바라 (Kolubara)	몽클로아아라바카 (Moncloa-Aravaca)
	Jamo-advanced	**나프탈렌** **(Naphthalene)**	테레프탈산 (Terephthalic acid)	아디프산 (Adipic acid)
스테이크 **(Steak)** **OOV word**	Baseline	스태너프 (Stanhope)	스태너드 (Stannard)	화이트스네이크 (White Snake)
	Jamo-advanced	롱테이크 (Long take)	비프스테이크 (Beefsteak)	**스테이크** **(Steak)**

6 Conclusions and Further Works

We introduce word representations specialized in Korean using morpheme analysis and extra sub-characters (Jamo). The similarity test shows higher correlation than the baseline model; thus, our model is competitive for word representations with semantic relations. For misspelled OOV words, our model shows a significantly increased cosine similarity between misspelled words and corrected words. Moreover, among the 10 most similar misspelled OOV words, the corrected words are included almost 80% of the time. Therefore, our model successfully solved the OOV problem caused by misspellings in Korean. Furthermore, it is very promising that our model can be utilized as a spelling correction method. We will try to apply our model for spelling correction and utilize it as an input for NLP applications, such as NER or MRC. By applying our model, we expect that the performance of NLP applications will be increased.

Acknowledgment

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (2013-0-00109, WiseKB: Big data based self-evolving knowledge base and reasoning platform).

References

1. T. Mikolov, K. Chen, G. Corrado, and J. Dean: Efficient estimation of word representations in vector space, *arXiv preprint arXiv:1301.3781*, (2013).
2. J. Pennington, R. Socher, and C. Manning: Glove: Global vectors for word representation, In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532-1543, (2014).
3. S. K. Sienčnik: Adapting word2vec to named entity recognition, In: Proceedings of the 20th nordic conference of computational linguistics, nodalida 2015, may 11-13, 2015, vilnius, lithuania, pp. 239-243, Linköping University Electronic Press, (2015).
4. M. Hu, Y. Peng, and X. Qiu: Reinforced mnemonic reader for machine comprehension, *CoRR, abs/1705.02798*, (2017).
5. J. Wieting, M. Bansal, K. Gimpel, and K. Livescu: Charagram: Embedding words and sentences via character n-grams, *arXiv preprint arXiv:1607.02789*, (2016).
6. P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov: Enriching word vectors with subword information, *arXiv preprint arXiv:1607.04606*, (2016).
7. G. Sampson:Writing systems, London,(1985).
8. H. Choi, H. Kwon, and A. Yoon: Improving Recall for Context-Sensitive Spelling Correction Rules using Conditional Probability Model with Dynamic Window Sizes, *Journal of KIISE,* vol. 42, no. 5, pp. 629-636, (2015).
9. S.-S. Kang, and Y. T. Kim: Syllable-based model for the Korean morphology, In: Proceedings of the 15th conference on Computational linguistics-Volume 1, pp. 221-226, Association for Computational Linguistics, (1994).
10. K. Stratos: A Sub-Character Architecture for Korean Language Processing, *arXiv preprint arXiv:1707.06341*, (2017).
11. T. Luong, R. Socher, and C. Manning: Better word representations with recursive neural networks for morphology, In: Proceedings of the Seventeenth Conference on Computational Natural Language Learning, pp. 104-113, (2013).
12. J. Botha, and P. Blunsom: Compositional morphology for word representations and language modelling, In: International Conference on Machine Learning, pp. 1899-1907, (2014).
13. T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean: Distributed representations of words and phrases and their compositionality, In: Advances in neural information processing systems, pp. 3111-3119, (2013).
14. A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov: Bag of tricks for efficient text classification, *arXiv preprint arXiv:1607.01759*, (2016).
15. A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou, and T. Mikolov: Fasttext. zip: Compressing text classification models, *arXiv preprint arXiv:1612.03651*, (2016).
16. L. Finkelstein, E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman, and E. Ruppin: Placing search in context: The concept revisited, In: Proceedings of the 10th international conference on World Wide Web, pp. 406-414, ACM, (2001).