한국어를 위한 개선된 워드 임베딩

츠나렐 재이다 ○ 장병탁
서울대학교
ceyda@bi.snu.ac.kr, btzhang@bi.snu.ac.kr

Better Word Embeddings for Korean

Ceyda Cinarel ○ Byoung-Tak Zhang
Seoul National University

Abstract

Vector representations of words that capture semantic and syntactic information accurately is critical for the performance of models that use these vectors as inputs. Algorithms that only use the surrounding context at the word level ignore the subword level relationships which carry important meaning especially for languages that are highly inflected such as Korean. In this paper we compare the word vectors generated by incorporating different levels of subword information, through visualization using t-SNE, for small sized Korean data.

1. Introduction

In order to use words as input to a neural network they have to be converted to vectors. One can simply use one-hot encoding where the vector size is equal to the vocabulary size and only the index of the corresponding word is 1 and rest is 0. But one-hot encoding doesn't inform the network about any linguistic meaning of the words. Embedding vectors that carry more relevant information about the language's structure and semantics can be obtained by using neural networks [4]. Architectures such as continuous bag of words (CBOW) or skip-gram use the surrounding context to create these embeddings. In sequence to sequence architecture the encoder RNN plays the word embedding vector creating role.

These methods that only use the word level to create the word embedding vectors may fail to capture syntactic relations. Especially for a language like Korean that have many words which are composed of meaningful blocks like hanja.

2. Related Research

The skip gram model can be extended to also use the subword information, as described in [1], by using a scoring function that sums over the set $\mathrm{G_w}$.

$$\mathrm{s(word, context)} = \sum_{g \in G_{word}} z_g^T v_{context} \quad (1)$$

Where $\mathrm{G_w}$ is a set of character n-grams (vectors) that comprises the word and the word itself. It is important to note that this results in changes in the vectors of words even when that word is not in the context window of the current target.

Another approach uses bidirectional LSTM to generate word embeddings from character level inputs (C2W) [6].

$$w^{char} = \mathrm{W^{forward}} h_w^{forward} + W^{reverse} h_w^{reverse} + \mathrm{bias} \quad (2)$$

These character derived embeddings $\mathrm{w^{ch}}$ then can be used together with word embeddings $\mathrm{w^{word}}$ by simply concatenating them together like;

$$\mathrm{w} = [w^{word}; w^{char}] \quad (3)$$

A gating mechanism can be used as introduced in [2]

$$\mathrm{g_w} = \sigma(v_g^T x_w^{word} + b_g) \quad (4)$$

Where σ is a sigmoid function. The gating function $\mathrm{g_w}$ learns to choose when to use word and when to use character derived vector embeddings

$$\mathrm{w} = (1 - g_w) w^{word} + \mathrm{g_w} w^{char} \quad (5)$$

2. Experiments

In order to compare how these methods work for Korean language the open source code* related to the researches described above was used. For Korean when we say character we mean a syllable i.e 안녕; 안+녕 (1-ch gram), 안녕 (2-ch gram)

2.1 Character Extended Skip Gram

A dataset containing 615449 sentences, 2594427 words from Korean drama scripts and subtitles was used.

* https://github.com/nyu-dl/gated_word_char_rlm
* https://github.com/facebookresearch/fastText

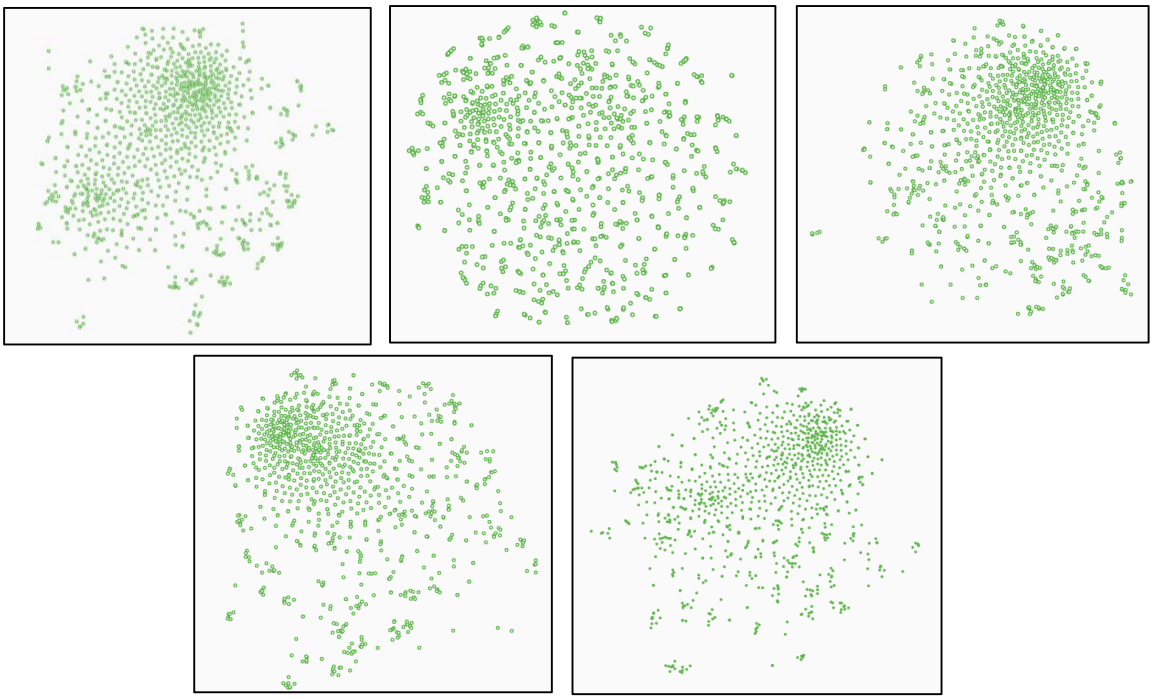

Figure 1 – Models left to right; word only, word+1ch, word+2ch, word+1ch+2ch, word+1ch+2ch+3ch.

The method described in [1] was used for five different 'word & character n-gram' combinations and the resulting vectors of the most frequent 1000 words were visualized using the dimensionality reduction technique t-SNE [3]. The combinations used in generating the set G_w for a word were; word only (baseline), word+1ch, word+2ch,word+2ch,word+1ch+2ch,word+1ch+2ch+3ch.

The word only model captured semantic relationships as can be seen in Figure 2. But overall the word vectors were still scattered.

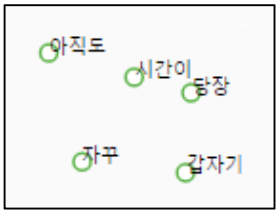

Figure 2 – from word only

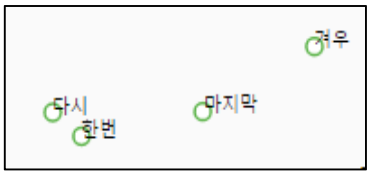

Figure 3 – word+ch1+ch2+ch3

Figure 4 – from word+1ch+2ch

Although words with different postpositional particles were better represented (Figure 4) in models using character n-grams, there were some unrelated words that were mapped to the same space because they

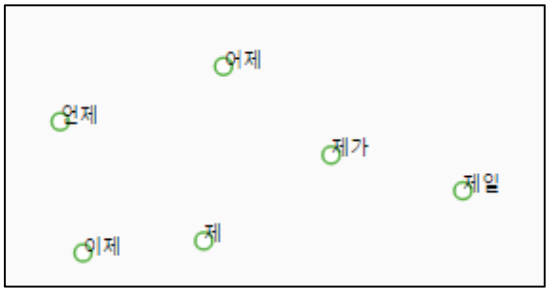

Figure 5 – from word+1ch

shared the same characters, can be seen in Figure 5. Especially word+1ch gram model was heavily influenced by the common syllables shared and mostly failed to capture semantic relations where there were no common syllables. Models that used combinations of more levels of n-grams, such as word+1ch+2ch, were able to preserve more of the semantic relations, Figure 3.

2.1 Gated Word Character

Four methods of generating word vectors as given in equations were used; C2W eq. (2), Concat eq. (3), Gated eq. (4)(5). Perplexity was calculated on the LSTM language model that predicts the next word given all the

previous words using one of these different word embeddings as the input. LSTM language model's hyperparameters were fixed and an embedding vector dimension of 100 was used for all the models to allow the results to be comparable.
As the dataset, collection of mother-child conversations in Korean that consists of ~20000 sentences and 74568 words was used. The dataset was shuffled and divided into training, validation and test sets with ratios 0.8, 0.1, 0.1 respectively.

	Word (baseline)	C2W	Concat (Word;C2W)	Gated
Validation	82	74	77	78
Test	91	84	87	89

Table 1. Perplexity calculated for the four different word embeddings, lower is better.

These results show that the models that also use the character level information were better than or at least as good as the word only baseline. A larger dataset might be necessary for the gating function to learn better.

3. Discussion and Future Research Direction

From looking at the results of the first experiment we can say that using word & multiple levels of character grams capture more syntactic relations.
Further improvements can be attempted by only choosing to include n-grams that have semantic or syntactic meaning in the set G_w. Since not all syllables have meaning or a syllable can have multiple different meanings. Similar to the role of gate in (5) the sum over the set in (1) can be replaced by a weighted sum, where the weights are hyper parameters.
These are just initial results that show optimistic potential and a more in-depth quantitative evaluation of these methods can reveal more.To better evaluate these word embeddings creating an analogy task set for Korean is necessary. There are also other considerations such as speed of these different methods as datasets get larger.

References

[1] Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov. "Enriching Word Vectors with Subword Information." arXiv preprint arXiv:1607.04606 (2016)
[2] Miyamoto, Yasumasa, and Kyunghyun Cho. "Gated Word-Character Recurrent Language Model." arXiv preprint arXiv:1606.01700 (2016)
[3] Maaten, Laurens van der, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of Machine Learning Research 9, pp.2579-2605 no. Nov (2008).
[4] Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S. Corrado, and Jeff Dean. "Distributed representations of words and phrases and their compositionality." In Advances in neural information processing systems, pp. 3111-3119. (2013)
[5] Chen, Xinxiong, Lei Xu, Zhiyuan Liu, Maosong Sun, and Huanbo Luan. "Joint learning of character and word embeddings." In Proceedings of IJCAI, pp. 1236-1242. (2015)
[6] Ling, Wang, Tiago Luís, Luís Marujo, Ramón Fernandez Astudillo, Silvio Amir, Chris Dyer, Alan W. Black, and Isabel Trancoso. "Finding function in form: Compositional character models for open vocabulary word representation." arXiv preprint arXiv:1508.02096 (2015)