Word-like character n-gram embedding

Geewook Kim and **Kazuki Fukui** and **Hidetoshi Shimodaira**
Department of Systems Science, Graduate School of Informatics, Kyoto University
Mathematical Statistics Team, RIKEN Center for Advanced Intelligence Project
{geewook, k.fukui}@sys.i.kyoto-u.ac.jp, shimo@i.kyoto-u.ac.jp

Abstract

We propose a new word embedding method called *word-like character n-gram embedding*, which learns distributed representations of words by embedding word-like character n-grams. Our method is an extension of recently proposed *segmentation-free word embedding*, which directly embeds frequent character n-grams from a raw corpus. However, its n-gram vocabulary tends to contain too many non-word n-grams. We solved this problem by introducing an idea of *expected word frequency*. Compared to the previously proposed methods, our method can embed more words, along with the words that are not included in a given basic word dictionary. Since our method does not rely on word segmentation with rich word dictionaries, it is especially effective when the text in the corpus is in unsegmented language and contains many neologisms and informal words (e.g., Chinese SNS dataset). Our experimental results on Sina Weibo (a Chinese microblog service) and Twitter show that the proposed method can embed more words and improve the performance of downstream tasks.

1 Introduction

Most existing word embedding methods require word segmentation as a preprocessing step (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017). The raw corpus is first converted into a sequence of words, and word co-occurrence in the segmented corpus is used to compute word vectors. This conventional method is referred to as Segmented character N-gram Embedding (SNE) for making a distinction clear in the argument below. Word segmentation is almost obvious for segmented languages (e.g., English), whose words are delimited by spaces. On the other hand, when dealing with unsegmented languages (e.g., Chinese and Japanese), whose word boundaries are not obviously indicated, word segmentation tools are used to determine word boundaries in the raw corpus. However, these segmenters require rich dictionaries for accurate segmentation, which are expensive to prepare and not always available. Furthermore, when we deal with noisy texts (e.g., SNS data), which contain a lot of neologisms and informal words, using a word segmenter with a poor word dictionary results in significant segmentation errors, leading to degradation of the quality of learned word embeddings.

Table 1: Top-10 2-grams in Sina Weibo and 4-grams in Japanese Twitter (Experiment 1). Words are indicated by boldface and space characters are marked by ␣.

	FNE		WNE (Proposed)	
	Chinese	Japanese	Chinese	Japanese
1	][	wwww	**自己**	**フォロー**
2	。␣	！！！！	。␣	**ありがと**
3	！␣	**ありがと**	][	wwww
4	..	りがとう	**一个**	！！！！
5	]␣	ございま	**微博**	**めっちゃ**
6	。。	うござい	**什么**	んだけど
7	，我	とうござ	**可以**	うござい
8	！！	ざいます	**没有**	**line**
9	␣我	がとうご	吗？	**2018**
10	了，	**ください**	**哈哈**	じゃない

To avoid the difficulty, *segmentation-free word embedding* has been proposed (Oshikiri, 2017). It does not require word segmentation as a preprocessing step. Instead, it examines frequencies of all possible character n-grams in a given corpus to build up *frequent n-gram lattice*. Subsequently, it composes distributed representations of n-grams by feeding their co-occurrence information to existing word embedding models. In this method, which we refer to as Frequent character N-gram Embedding (FNE), the top-K most frequent character n-grams are selected as n-gram vocabulary for embedding. Although FNE does not require any word dictionaries, the n-gram vocabulary tends to include a vast amount of non-words. For example, only 1.5% of the n-gram vocabulary is estimated as words at $K = 2M$ in Experiment 1 (See Precision of FNE in Fig. 2b).

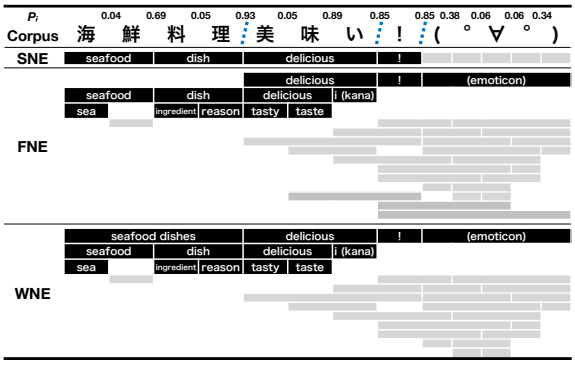

Figure 1: A Japanese tweet with manual segmentation. The output of a standard Japanese word segmenter[4] is shown in SNE. The n-grams included in the vocabularies of each method are shown in FNE and WNE ($K=2\times10^6$). Words are black and non-words are gray.

Since the vocabulary size K is limited, we would like to reduce the number of non-words in the vocabulary in order to embed more words. To this end, we propose another segmentation-free word embedding method, called *Word-like character N-gram Embedding* (WNE). While FNE only considers n-gram frequencies for constructing the n-gram vocabulary, WNE considers how likely each n-gram is a "word". Specifically, we introduce the idea of *expected word frequency* (***ewf***) in a stochastically segmented corpus (Mori and Takuma, 2004), and the top-K n-grams with the highest **ewf** are selected as n-gram vocabulary for embedding. In WNE, **ewf** estimates the frequency of each n-gram appearing as a word in the corpus, while the raw frequency of the n-gram is used in FNE. As seen in Table 1 and Fig. 1, WNE tends to include more dictionary words than FNE.

WNE incorporates the advantage of dictionary-based SNE into FNE. In the calculation of **ewf**, we use a probabilistic predictor of word boundary. We do not expect the predictor is very accurate—If it is good, SNE is preferred in the first place. A naive predictor is sufficient for giving low **ewf** score to the vast majority of non-words so that words, including neologisms, are easier to enter the vocabulary. Although our idea seems somewhat simple, our experiments show that WNE significantly improves word coverage while achieving better performances on downstream tasks.

2 Related work

The lack of word boundary information in unsegmented languages, such as Chinese and Japanese, raises the need for an additional step of word segmentation, which requires rich word dictionaries to deal with corpora consisting of a lot of neologisms. However, in many cases, such dictionaries are costly to obtain or to maintain up-to-date. Though recent studies have employed character-based methods to deal with large size vocabulary for NLP tasks ranging from machine translation (Costa-jussà and Fonollosa, 2016; Luong and Manning, 2016) to part-of-speech tagging (Dos Santos and Zadrozny, 2014), they still require a segmentation step. Some other studies employed character-level or n-gram embedding without word segmentation (Schütze, 2017; Dhingra et al., 2016), but most cases are task-specific and do not set their goal as obtaining word vectors. As for word embedding tasks, subword (or n-gram) embedding techniques have been proposed to deal with morphologically rich languages (Bojanowski et al., 2017) or to obtain fast and simple architectures for word and sentence representations (Wieting et al., 2016), but these methods do not consider a situation where word boundaries are missing. To obtain word vectors without word segmentation, Oshikiri (2017) proposed a new pipeline of word embedding which is effective for unsegmented languages.

3 Frequent n-gram embedding

A new pipeline of word embedding for unsegmented languages, referred to as FNE in this paper, has been proposed recently in Oshikiri (2017). First, the frequencies of all character n-grams in a raw corpus are counted for selecting the K-most frequent n-grams as the n-gram vocabulary in FNE. This way of determining n-gram vocabulary can also be found in Wieting et al. (2016). Then frequent n-gram lattice is constructed by enumerating all possible segmentations with the n-grams in the vocabulary, allowing partial overlapping of n-grams in the lattice. For example, assuming that there is a string "短い学術論文" (short academic paper) in a corpus, and if 短い (short), 学術 (academic), 論文 (paper) and 学術論文 (academic paper) are included in the n-gram vocabulary, then word and context pairs are (短い, 学術), (短い, 学術論文) and (学術, 論文). Co-occurrence frequencies over the frequent n-gram lattice are fed into the word embedding model to obtain vectors of n-grams in the vocabulary. Consequently, FNE succeeds to learn embeddings for many words while avoiding the negative impact of the erroneous segmentations.

Although FNE is effective for unsegmented languages, it tends to embed too many non-words. This is undesirable since the number of embedding targets is limited due to the time and memory constraints, and the non-words in the vocabulary could degrade the quality of the word embeddings.

4 Word-like n-gram embedding

To reduce the number of non-words in the n-gram vocabulary of FNE, we change the selection criterion of n-grams. In FNE, the selection criterion of a given n-gram is its frequency in the corpus. In our proposal WNE, we replace the frequency with the *expected word frequency* (***ewf***). **ewf** is the expected frequency of a character n-gram appearing as a word over the corpus by taking account of context information. For instance, given an input string "美容院でカラーリングする" (Do hair coloring at a beauty shop), FNE simply counts the occurrence frequency of リング (ring) and ignores the fact that it breaks the meaning of カラーリング (coloring), whereas **ewf** suppresses the counting of リング by evaluating how likely the リング appeared as a word in the context. **ewf** is called as stochastic frequency in Mori and Takuma (2004).

4.1 Expected word frequency

Mori and Takuma (2004) considered the stochastically segmented corpus with probabilistic word boundaries. Let $x_1 x_2 \cdots x_N$ be a raw corpus of N characters, and Z_i be the indicator variable for the word boundary between two characters x_i and x_{i+1}; $Z_i = 1$ when the boundary exists and $Z_i = 0$ otherwise. The word boundary probability is denoted by $P(Z_i = 1) = P_i$ and $P(Z_i = 0) = 1 - P_i$, where P_i is calculated from the context as discussed in Section 4.2.

Here we explain **ewf** for a character n-gram w by assuming that the sequence of word boundary probabilities $\mathbf{P}_0^N = (P_0, P_1, \cdots, P_N)$ is already at hand. Let us consider an appearance of the specified n-gram w in the corpus as $x_i x_{i+1} \cdots x_j = w$ with length $n = j - i + 1$. The set of all such appearances is denoted as $I(w) = \{(i, j) \mid x_i x_{i+1} \cdots x_j = w\}$. By considering a naive independence model, the probability of $x_i x_{i+1} \cdots x_j$ being a word is $P(i, j) = P_{i-1} P_j \prod_{k=i}^{j-1} (1 - P_k)$, and **ewf** is simply the sum of $P(i, j)$ over the whole corpus

$$\mathbf{ewf}(w) = \sum_{(i,j) \in I(w)} P(i, j),$$

while the raw frequency of w is expressed as

$$\mathbf{freq}(w) = \sum_{(i,j) \in I(w)} 1.$$

4.2 Probabilistic predictor of word boundary

In this paper, a logistic regression is used for estimating word boundary probability. For explanatory variables, we employ the *association strength* (Sproat and Shih, 1990) of character n-grams; similar statistics of word n-grams are used in Mikolov et al. (2013) to detect phrases. The association strength of a pair of two character n-grams a, b is defined as

$$A(a, b) = \log\left(\frac{\mathbf{freq}(ab)}{N}\right) - \log\left(\frac{\mathbf{freq}(a)\mathbf{freq}(b)}{N^2}\right).$$

For a specified window size s, all the combinations of $a \in \{x_i, x_{i-1}x_i, \ldots, x_{i-s+1} \cdots x_i\}$ and $b \in \{x_{i+1}, x_{i+1}x_{i+2}, \ldots, x_{i+1} \cdots x_{i+s}\}$ are considered for estimating P_i.

5 Experiments

We evaluate the three methods: SNE, FNE and WNE. We use 100MB of SNS data, Sina Weibo[1] for Chinese and Twitter[2] for Japanese and Korean, as training corpora. Although Korean has spacing, the word boundaries are not obviously determined by space. The implementation of the proposed method is available on GitHub[3].

5.1 Comparison word embedding models

The three methods are combined with Skip-gram model with Negative Sampling (SGNS) (Mikolov et al., 2013), where the dimension of word embeddings is 200 and the number of epochs is 20. The initial learning rate γ and the number of negative samples n_{neg} are grid searched over $(\gamma, n_{\text{neg}}) \in \{0.01, 0.025\} \times \{5, 10\}$. The context window size h is grid searched over $h \in \{1, 5, 10\}$ in SNE, and $h = 1$ is used for FNE and WNE.

SGNS-SNE (baseline): The standard word segmenters[4] are used to obtain segmented corpora.

SGNS-FNE (baseline): SGNS is extended to allow frequent n-gram lattice in Oshikiri (2017). In

[1] We used 100MB of Leiden Weibo Corpus (van Esch, 2012) from the head.

[2] We collected Japanese and Korean tweets using the Twitter Streaming API.

[3] https://github.com/kdrl/WNE

[4] MeCab with IPADIC is used for Japanese, jieba with jieba/dict.txt.small are used for Chinese, and MeCab-ko with mecab-ko-dic is used for Korean.

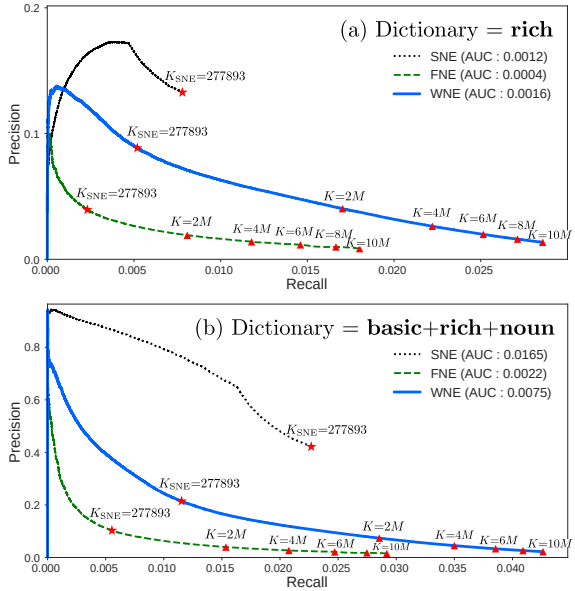

Figure 2: Precision-Recall curves for Japanese in two sets of dictionaries (Experiment 1). The maximum K of SNE (K_{SNE}) is indicated by star.

this model, the n-gram vocabulary is constructed with the K-most frequent n-grams and the embeddings of n-grams are computed by utilizing its co-occurrence information over the frequent n-gram lattice.

SGNS-WNE (Proposed model): We modified SGNS-FNE by replacing the n-gram frequency with **ewf**. To estimate word boundary probabilities, the logistic regression of window size $s = 8$ is trained with randomly sampled 1% of the corpus segmented by the same basic word segmenters[4] used in SNE. Again, we do not expect here the probabilistic predictor of word boundary is very accurate. A naive predictor is sufficient for giving low **ewf** score to the vast majority of non-words.

5.2 Experiment 1: Selection criteria of embedding targets

We examine the number of words and non-words in the n-gram vocabulary. The n-gram vocabularies of size K are prepared by the three methods. For evaluating the vocabularies, we prepared three types of dictionaries for each language, namely, **basic**, **rich**[5] and **noun**. **basic** is the standard dictionary for the word segmenters, and **rich** is a larger dictionary including neologisms. **noun** is a word set consists of all noun words in Wikidata (Vrandečić and Krötzsch, 2014).

Each n-gram in a vocabulary is marked as "word" if it is included in a specified dictionary. We then compute Precision as the ratio of marked words in the vocabulary and Recall as the ratio of marked words in the dictionary. Precision-Recall curve is drawn by changing K from 1 to 1×10^7.

[5]For Japanese, Chinese, and Korean, respectively, basic dictionaries are IPADIC, jieba/dict.txt.small, mecab-ko-dic, and rich dictionaries are NEologd, jieba/dict.txt.big, NIADic

Table 2: Classification accuracies [%] (Experiment 2)

Model	Lang.	**Recall**[a]	**Acc$_{\text{uni}}$**[b]	Acc$_{\text{int}}$[c]
SGNS-SNE		18.07	61.31	81.19
SGNS-FNE	Chinese	11.36	35.61	86.44
SGNS-WNE		**20.68**	**73.64**	**87.23**
SGNS-SNE		0.78	44.50	79.56
SGNS-FNE	Japanese	0.81	39.06	80.50
SGNS-WNE		**1.70**	**69.76**	**81.70**
SGNS-SNE		7.36	62.51	77.35
SGNS-FNE	Korean	4.21	43.87	84.30
SGNS-WNE		**9.38**	**74.50**	**84.32**

[a] Dictionary = **rich**, [b] Union of the three vocabularies, [c] Intersection of the three vocabularies.

5.3 Experiment 2: Noun category prediction

We performed the noun category prediction task with the learned word vectors. Most of the settings are the same as Oshikiri (2017). Noun words and their categories are extracted from Wikidata with the predetermined category set[6]. The word set is split into train (60%) and test (40%) sets. The hyperparameters are tuned with 5-folds CV on the train set, and the performance is measured on the test set. This is repeated 10 times for random splits, and the mean accuracies are reported. C-SVM classifiers are trained to predict categories from the word vectors, where unseen words are skipped in training and treated as errors in testing. We performed a grid search over $(C, \text{classifier}) \in \{0.5, 1, 5, 10, 50\} \times \{\text{1-vs.-1}, \text{1-vs.-all}\}$ of linear SVM. The vocabulary size is set to $K = 2 \times 10^6$ for FNE and WNE, while $K = K_{\text{SNE}}$ is fixed at the maximum value, i.e., the number of unique segmented n-grams for SNE.

5.4 Result

The results of experiments are shown in Fig. 2 and Table 2. PR-curves for Chinese and Korean are similar to Japanese and omitted here. As expected, SNE has the highest Precision. WNE improves Precision of FNE greatly by reducing non-words in the vocabulary. On the other hand, WNE has the highest Recall (the coverage of dictionary words) for large K, followed by FNE. Since SNE cannot increase K beyond K_{SNE}, its Recall is limited.

[6]{human, fictional character, manga, movie, girl group, television drama, year, company, occupation, color, country}

Looking at the classification accuracies computed for the intersection of the vocabularies of SNE, FNE and WNE, they are relatively similar, while looking at those for the union of the vocabularies, WNE is the highest. This indicates that the quality of the word vectors is similar in the three methods, but the high coverage of WNE contributes to the performance improvement of the downstream task compared to SNE and FNE.

6 Conclusion

We proposed WNE, which trains embeddings for word-like character n-grams instead of segmented n-grams. Compared to the other methods, the proposed method can embed more words, along with the words that are not included in the given word dictionary. Our experimental results show that WNE can learn high-quality representations of many words, including neologisms, informal words and even text emoticons. This improvement is highly effective in real-world situations, such as dealing with large-scale SNS data. The other word embedding models, such as FastText (Bojanowski et al., 2017) and GloVe (Pennington et al., 2014), can also be extended with WNE.

Acknowledgments

We would like to thank Tetsuya Hada, Shinsuke Mori and anonymous reviewers for their helpful advices. This work was partially supported by JSPS KAKENHI grant 16H02789 to HS and 18J15053 to KF.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association of Computational Linguistics*, 5:135–146.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361. Association for Computational Linguistics.

Bhuwan Dhingra, Zhong Zhou, Dylan Fitzpatrick, Michael Muehl, and William Cohen. 2016. Tweet2vec: Character-based distributed representations for social media. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 269–274, Berlin, Germany. Association for Computational Linguistics.

Cícero Nogueira Dos Santos and Bianca Zadrozny. 2014. Learning character-level representations for part-of-speech tagging. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pages II–1818–II–1826. JMLR.org.

Daan van Esch. 2012. Leidon weibo corpus. Http://lwc.daanvanesch.nl.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1054–1063, Berlin, Germany. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Shinsuke Mori and Daisuke Takuma. 2004. Word n-gram probability estimation from a japanese raw corpus. In *Eighth International Conference on Spoken Language Processing*.

Takamasa Oshikiri. 2017. Segmentation-free word embedding for unsegmented languages. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 767–772, Copenhagen, Denmark. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, volume 14, pages 1532–1543.

Hinrich Schütze. 2017. Nonsymbolic text representation. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 785–796. Association for Computational Linguistics.

Richard Sproat and Chilin Shih. 1990. *A Statistical Method for Finding Word Boundaries in Chinese Text*, volume 4. Computer Processing of Chinese and Oriental Languages.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2016. Charagram: Embedding words and sentences via character n-grams. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1515. Association for Computational Linguistics.