Segmentation-free Compositional 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 type of representation learning method that models words, phrases and sentences seamlessly. Our method does not depend on word segmentation and any human-annotated resources (e.g., word dictionaries), yet it is very effective for noisy corpora written in unsegmented languages such as Chinese and Japanese. The main idea of our method is to ignore word boundaries completely (i.e., *segmentation-free*), and construct representations for all character n-grams in a raw corpus with embeddings of *compositional* sub-n-grams. Although the idea is simple, our experiments on various benchmarks and real-world datasets show the efficacy of our proposal.

1 Introduction

Most existing word embedding models (Mikolov et al., 2013; Pennington et al., 2014; Bojanowski et al., 2017) take a sequence of words as their input. Therefore, the conventional models are dependent on word segmentation (Yang et al., 2017; Shao et al., 2018), which is a process of converting a raw corpus (i.e., a sequence of characters) into a sequence of segmented character n-grams. After the segmentation, the segmented character n-grams are assumed to be words, and each word's representation is constructed from distribution of neighbour words that co-occur together across the estimated word boundaries. However, in practice, this kind of approach has several problems. First, word segmentation is difficult especially when texts in a corpus are noisy or unsegmented (Saito et al., 2014; Kim et al., 2018). For example, word segmentation on social network service (SNS) corpora, such as Twitter, is a challenging task since it tends to include many misspellings, informal words, neologisms, and even emoticons. This problem becomes more severe in unsegmented languages, such as Chinese and Japanese, whose word boundaries are not explicitly indicated. Second, word segmentation has ambiguities (Luo et al., 2002; Li et al., 2003). For example, a compound word 線形代数学 (linear algebra) can be seen as a single word or sequence of words, such as 線形|代数学 (linear | algebra).

Word segmentation errors negatively influence subsequent processes (Xu et al., 2004). For example, we may lose some words in training corpora, leading to a larger Out-Of-Vocabulary (OOV) rate (Sun et al., 2005). Moreover, segmentation errors, such as segmenting きのう (yesterday) as き|のう (tree | brain), produce false co-occurrence information. This problem is crucial for most existing word embedding methods as they are based on distributional hypothesis (Harris, 1954), which can be summarized as: "*a word is characterized by the company it keeps*" (Firth, 1957).

To enhance word segmentation, some recent works (Junyi, 2013; Sato, 2015; Jeon, 2016) made rich resources publicly available. However, maintaining them up-to-date is difficult and it is infeasible for them to cover all types of words. To avoid the negative impacts of word segmentation errors, Oshikiri (2017) proposed a word embedding method called *segmentation-free word embedding* (`sembei`). The key idea of `sembei` is to directly embed frequent character n-grams from a raw corpus without conducting word segmentation. However, most of the frequent n-grams are non-words (Kim et al., 2018), and hence `sembei` still suffers from the OOV problems. The fundamental problem also lies in its extension (Kim et al., 2018), although it uses external resources to reduce the number of OOV. To handle OOV problems, Bojanowski et al. (2017) proposed a novel *compositional* word embedding method with subword modeling, called *subword-information skip-gram* (`sisg`). The key idea of `sisg` is to extend the notion of vocabulary to include subwords,

Proceedings of NAACL-HLT 2019, pages 3207–3215
Minneapolis, Minnesota, June 2 - June 7, 2019.

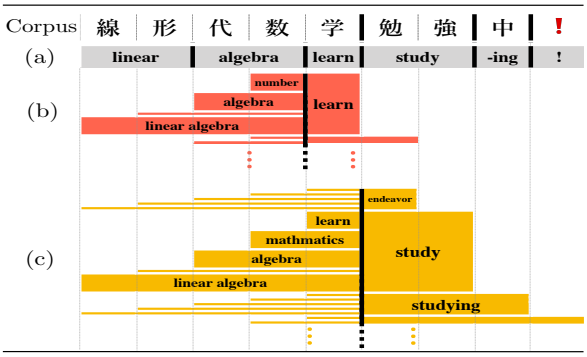

Figure 1: A Japanese tweet with manual segmentation. (a) is the segmentation result of a widely-used word segmenter which conventional word embedding methods are dependent on. (b) and (c) show the embedding targets and their co-occurrence information to be considered in our proposed method `scne` on the boundaries of 数|学 and 学|勉. Unlike conventional word embedding methods, `scne` considers all possible character n-grams on all boundaries (e.g., 線|形, 形|代, 代|数, $\cdots$) in the raw corpus without segmentation.

namely, substrings of words, for enriching the representations of words by the embeddings of its subwords. In `sisg`, the embeddings of OOV (or unseen) words are computed from the embedings of their subwords. However, `sisg` requires word segmentation as a prepossessing step, and the way of collecting co-occurrence information is dependent on the results of explicit word segmentation.

For solving the issues of word segmentation and OOV, we propose a simple but effective unsupervised representation learning method for words, phrases and sentences, called *segmentation-free compositional n-gram embedding* (`scne`). The key idea of `scne` is to train embeddings of character n-grams to compose representations of all character n-grams in a raw corpus, and it enables treating all words, phrases and sentences seamlessly (see Figure 1 for an illustrative explanation). Our experimental results on a range of datasets suggest that `scne` can compute high-quality representations for words and sentences although it does not consider any word boundaries and is not dependent on any human annotated resources.

2 Segmentation-free Compositional n-gram Embedding (`scne`)

Our method `scne` successfully combines a subword model (Zhang et al., 2015; Wieting et al., 2016; Bojanowski et al., 2017; Zhao et al., 2018) with an idea of character n-gram embedding (Oshikiri, 2017; Kim et al., 2018). In `scne`, the vector representation of a target character n-gram is defined as follows. Let $x_1 x_2 \cdots x_N$ be a raw unsegmented corpus of N characters. For a range $i, i+1, \ldots, j$ specified by index $t = (i, j)$, $1 \leq i \leq j \leq N$, we denote the substring $x_i x_{i+1} \cdots x_j$ as $\boldsymbol{x}_{(i,j)}$ or $\boldsymbol{x}_t$. In a training phase, `scne` first counts frequency of character n-grams in the raw corpus to construct n-gram set V by collecting M-most frequent n-grams with $n \leq n_{\max}$, where M and $n_{\max}$ are hyperparameters. For any target character n-gram $\boldsymbol{x}_{(i,j)} = x_i x_{i+1} \cdots x_j$ in the corpus, `scne` constructs its representation $v_{\boldsymbol{x}_{(i,j)}} \in \mathbb{R}^d$ by summing the embeddings of its sub-n-grams as follows:

$$v_{\boldsymbol{x}_{(i,j)}} = \sum_{s \in S(\boldsymbol{x}_{(i,j)})} z_s,$$

where $S(\boldsymbol{x}_{(i,j)}) = \{\boldsymbol{x}_{(i',j')} \in V \mid i \leq i' \leq j' \leq j\}$ consists of all sub-n-grams of target $\boldsymbol{x}_{(i,j)}$, and the embeddings of sub-n-grams $z_s \in \mathbb{R}^d$, $s \in V$ are model parameters to be learned. The objective of `scne` is similar to that of Mikolov et al. (2013),

$$\sum_{t \in \mathcal{D}} \left\{ \sum_{c \in \mathcal{C}(t)} \log \sigma\left(v_{\boldsymbol{x}_t}^\top u_{\boldsymbol{x}_c}\right) + \sum_{\tilde{s} \sim P_{\text{neg}}}^{k} \log \sigma\left(-v_{\boldsymbol{x}_t}^\top u_{\tilde{s}}\right) \right\},$$

where $\sigma(x) = \frac{1}{1+\exp(-x)}$, $\mathcal{D} = \{(i,j) \mid 1 \leq i \leq j \leq N, j - i + 1 \leq n_{\text{target}}\}$, and $\mathcal{C}((i,j)) = \{(i',j') \mid \boldsymbol{x}_{(i',j')} \in V, j' = i-1 \text{ or } i' = j+1\}$. $\mathcal{D}$ is the set of indexes of all possible target n-grams in the raw corpus with $n \leq n_{\text{target}}$, where n_{target} is a hyperparameter. $\mathcal{C}(t)$ is the set of indexes of contexts of the target $\boldsymbol{x}_t$, that is, all character n-grams in V that are adjacent to the target (see Figures 1 and 2). The negative sampling distribution P_{neg} of $\tilde{s} \in V$ is proportional to its frequency in the corpus. The model parameters $z_s, u_{\tilde{s}} \in \mathbb{R}^d$, $s, \tilde{s} \in V$, are learned by maximizing the objective. We set $n_{\text{target}} = n_{\max}$ in our experiments.

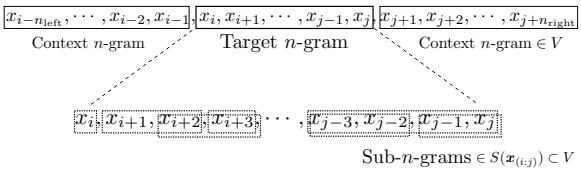

Figure 2: A graphical illustration of the proposed model trying to compute a representation for a character n-gram $\boldsymbol{x}_{(i,j)}$. The co-occurrence of $\boldsymbol{x}_{(i,j)}$ and its neighbouring context n-grams are used to train embeddings of *compositional* n-grams.

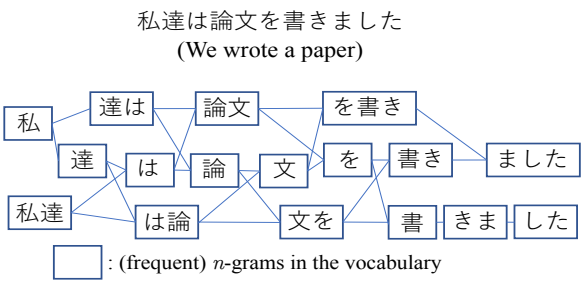

Figure 3: An example of a frequent n-gram lattice.

Although we examine frequent n-grams for simplicity, incorporating supervised word boundary information or *byte pair encoding* into the construction of *compositional* n-gram set would be an interesting future work (Kim et al., 2018; Sennrich et al., 2016; Heinzerling and Strube, 2018).

2.1 Comparison to Oshikiri (2017)

To avoid the problems of word segmentation, Oshikiri (2017) proposed *segmentation-free word embedding* (sembei) (Oshikiri, 2017) that considers the M-most frequent character n-grams as individual words. Then, a *frequent n-gram lattice* is constructed, which is similar to a word lattice used in morphological analysis (see Figure 3). Finally, the pairs of adjacent n-grams in the lattice are considered as target-context pairs and they are fed to existing word embedding methods, e.g., skipgram (Mikolov et al., 2013). Although sembei is simple, the frequent n-gram vocabulary tends to include a vast amount of non-words (Kim et al., 2018). Furthermore, its vocabulary size is limited to M, hence, sembei can not avoid the undesirable issue of OOV. The proposed scne avoids these problems by taking all possible character n-grams as embedding targets. Note that the target-context pairs of sembei are fully contained in those of scne (see Figure 1).

2.2 Comparison to Kim et al. (2018)

To overcome the problem of OOV in sembei, Kim et al. (2018) proposed an extension of sembei called *word-like n-gram embedding* (wne). In wne, the n-gram vocabulary is filtered to have more vaild words by taking advantage of a supervised probabilistic word segmenter. Although wne reduce the number of non-words, there is still the problem of OOV since its vocabulary size is limited. In addition, wne is dependent on word segmenter while scne does not.

2.3 Comparison to Bojanowski et al. (2017)

To deal with OOV words as well as rare words, Bojanowski et al. (2017) proposed *subword information skip-gram* (sisg) that enriches word embeddings with the representations of its subwords, i.e., sub-character n-grams of words. In sisg, a vector representation of a target word is encoded as the sum of the embeddings of its subwords. For instance, subwords of length $n = 3$ of the word *where* are extracted as <wh, whe, her, ere, re>, where "<","<" are special symbols added to the original word to represent its left and right word boundaries. Then, a vector representation of *where* is encoded as the sum of the embeddings of these subwords and that of the special sequence <where>, which corresponds to the original word itself. Although sisg is powerful, it requires the information of word boundaries as its input, that is, semantic units need to be specified when encoding targets. Therefore, it cannot be directly applied to unsegmented languages. Unlike sisg, scne does not require such information. The proposed scne is much simpler, but due to its simpleness, the embedding target of scne should contains many non-words, which seems to be a problem (see Figure 1). However, our experimental results show that scne successfully captures the semantics of words and even sentences for unsegmented languages without using any knowledge of word boundaries (see Section 3).

3 Experiments

In this section, we perform two intrinsic and two extrinsic tasks at both word and sentence level, focusing on unsegmented languages. The implementation of our method is available on GitHub[1].

3.1 Common Settings

Baselines: We use skipgram (Mikolov et al., 2013), sisg (Bojanowski et al., 2017) and sembei (Oshikiri, 2017) as word embedding baselines. For sentence embedding, we first test simple baselines obtained by averaging the word vectors over a word-segmented sentence. In addition, we examine several recent successful sentence embedding methods, pv-dbow, pv-dm (Le and Mikolov, 2014) and sent2vec (Pagliardini et al., 2018) in an extrinsic task. Note that both scne and sembei have embeddings of frequent character n-grams as their model parameters, but

[1] www.github.com/kdrl/SCNE

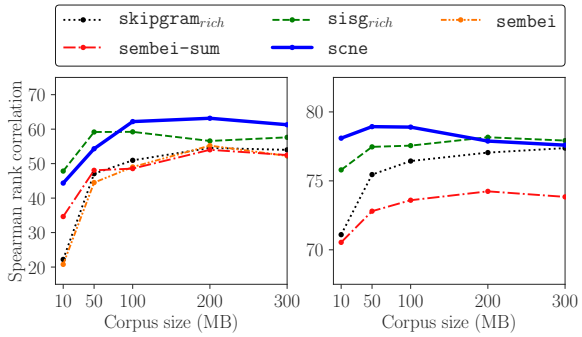

Figure 4: Word (left) and sentence (right) similarity tasks on portions of Chinese Wikipedia corpus.

the differences come from training strategies, such as embedding targets and the way of collecting co-occurrence information (see Section 2.1 for more details). For contrasting `scne` with `sembei`, we also propose a variant of `sembei` (denoted by `sembei-sum`) as one of baselines, which composes word and sentence embeddings by simply summing up the embeddings of their sub-n-grams which are learned by `sembei`.

Hyperparameters Tuning: To see the effect of rich resources for the segmentation-dependent baselines, we employ widely-used word segmenter with two settings: Using only a basic dictionary (*basic*) or using a rich dictionary together (*rich*). The dimension of embeddings is 200, the number of epochs is 10 and the number of negative samples is 10 for all the methods. The n-gram vocabulary size $M = 2 \times 10^6$ is used for `sisg`, `sembei` and `scne`. The other hyperparameters, such as learning rate and $n_{\max}$, are carefully adjusted via a grid search in the validation set. In the word similarity task, 2-fold cross validation is used for evaluation. In the sentence similarity task, we use the provided validation set. In the downstream tasks, vector representations are combined with a supervised logistic regression classifier. We repeat training and testing of the classifier 10 times, while the prepared dataset is randomly split into train (60%) and test (40%) sets at each time, and the hyperparameters are tuned by 3-fold cross validation in the train set. We adopt mean accuracy as the evaluation metric. See Appendix A.1 for more experimental details.

3.2 Word and Sentence Similarity

We measure the ability of models to capture semantic similarity for words and sentences in Chinese; see Appendix A.2 for the experiment in Japanese. Given a set of word pairs, or sentence pairs, and their human annotated similarity scores, we calculated Spearman's rank correlation between the cosine similarities of the embeddings and the scores. We use the dataset of Jin and Wu (2012) and Wang et al. (2017) for Chinese word and sentence similarity respectively. Note that the conventional models, such as `skipgram`, cannot provide the embeddings for OOV words, while the *compositional* models, such as `sisg` and `scne`, can compute the embeddings by using their subword modeling. In order to show comparable results, we use the null vector for these OOV words following Bojanowski et al. (2017).

	skipgram$_{rich}$	sisg$_{rich}$	sembei	sembei-sum	**scne**
Wiki.	51.0	<u>59.2</u>	49.0	48.6	**62.2**
SNS	41.3	<u>47.0</u>	38.9	41.5	**60.0**
Diff.	-9.7	-12.2	-10.1	-7.1	-2.2

Table 1: Spearman rank correlations of the word similarity task on two different Chinese corpora. Best scores are boldface and 2nd best scores are underlined.

Results: To see the effect of training corpus size, we train all models on portions of Wikipedia[2]. The results are shown in Figure 4. As it can be seen, the proposed `scne` is competitive with or outperforms the baselines for both word and sentence similarity tasks. Moreover, it is worth noting that `scne` provides high-quality representations even when the size of training corpus is small, which is crucial for practical real-world settings where rich data is not available. For a next experiment to see the effect of noisiness of training corpus, we test both noisy SNS corpus and the Wikipedia corpus[3] of the same size. The results are reported in Table 1. As it can be seen, the performance of segmentation-dependent methods (`skipgram`, `sisg`) are decreased greatly by the noisiness of the corpus, while `scne` degrades only marginally. The other two segmentation-free methods (`sembei`, `sembei-sum`) performed poorly. This shows the efficacy of our method in the noisy texts. On the other hand, in preliminary experiments on English (not shown), `scne` did not get better results than our segmentation-dependent baselines and it will be a future work to incorporate easily obtainable word boundary information into `scne` for segmented languages.

[2]We use 10, 50, 100, 300MB of Wikipedia from the head.
[3]We use 100MB of Sina Weibo posts for Chinese SNS corpus and 100MB of Chinese Wikipedia corpus.

	Wikipedia corpora						Noisy SNS corpora					
	Chinese		Japanese		Korean		Chinese		Japanese		Korean	
	All	Intersec.	**All**	Intersec.	**All**	Intersec.	**All**	Intersec.	**All**	Intersec.	**All**	Intersec.
skipgram$_{basic}$	8.9 (11)	81.0	7.8 (10)	75.7	11.5 (15)	**77.1**	2.9 (5)	58.2	2.9 (7)	41.4	2.4 (7)	35.4
skipgram$_{rich}$	9.5 (12)	81.0	16.7 (20)	**75.8**	11.9 (15)	76.9	3.0 (5)	58.2	4.1 (9)	40.9	2.5 (7)	34.2
sisg$_{basic}$	79.2 (100)	**82.3**	72.2 (100)	75.7	72.2 (100)	76.2	71.0 (100)	64.8	67.1 (100)	46.9	63.4 (100)	**39.8**
sisg$_{rich}$	79.5 (100)	82.2	73.3 (100)	74.7	72.4 (100)	76.6	70.8 (100)	64.9	67.5 (100)	46.0	63.3 (100)	37.7
sembei	21.8 (25)	79.0	18.2 (23)	70.1	14.2 (19)	41.8	4.5 (7)	59.6	4.9 (10)	41.9	5.0 (13)	33.7
sembei-sum	76.8 (100)	74.2	69.9 (100)	61.3	66.3 (100)	56.0	72.3 (100)	56.4	66.3 (100)	40.7	64.3 (100)	34.8
scne (Proposed)	**79.8 (100)**	81.5	**73.9 (100)**	74.0	**73.2 (100)**	73.9	**74.9 (100)**	**65.0**	**68.1 (100)**	**47.6**	**65.3 (100)**	38.2

Table 2: Noun category prediction accuracies (higher is better) and coverages [%] (in parentheses, higher is better).

	Segmentation-free	Chinese	Japanese	Korean
pv-dbow$_{basic}$		82.84	85.24	84.16
pv-dbow$_{rich}$		83.47	85.55	84.80
pv-dm$_{basic}$		76.96	80.67	66.35
pv-dm$_{rich}$		77.94	81.37	67.32
sent2vec$_{basic}$		85.09	87.12	82.31
sent2vec$_{rich}$		85.39	87.20	82.34
skipgram$_{basic}$		85.79	86.76	84.06
skipgram$_{rich}$		85.77	87.16	84.48
sisg$_{basic}$		85.67	87.22	84.34
sisg$_{rich}$		85.04	87.25	84.35
sembei-sum	✓	83.41	80.80	74.98
scne$_{n_{\max}=8}$	✓	87.07	87.42	84.15
scne$_{n_{\max}=16}$	✓	**87.76**	**88.03**	**86.74**

Table 3: Sentiment classification accuracies [%].

3.3 Noun Category Prediction

As a word-level downstream task, we conduct a noun category prediction on Chinese, Japanese and Korean[4]. Most settings are the same as those of Oshikiri (2017). Noun words and their semantic categories are extracted from Wikidata (Vrandečić and Krötzsch, 2014) with a predetermined semantic category set[5], and the classifier is trained to predict the semantic category of words from the learned word representations, where unseen words are skipped in training and treated as errors in testing. To see the effect of the noisiness of corpora, both noisy SNS corpus and Wikipedia corpus of the same size are examined as training corpora[6].

Results: The results are reported in Table 2. Since the set of covered nouns (i.e., non-OOV words) depends on the methods, we calculate accuracies in two ways for a fair comparison: Using all the nouns and using the intersection of the covered nouns. scne achieved the highest accuracies in all the settings when using all the nouns, and also performed well when using the intersection of the covered nouns, especially for the noisy corpora.

3.4 Sentiment Analysis

As a sentence-level evaluation, we perform sentiment analysis on movie review data. We use 101k, 56k and 200k movie reviews and their scores respectively from Chinese, Japanese and Korean movie review websites (see Appendix A.1.6 for more details). Each review is labeled as positive or negative by its rating score. Sentence embedding models are trained using the whole movie reviews as training corpus. Among the reviews, 5k positive and 5k negative reviews are randomly selected, and the selected reviews are used to train and test the classifiers as explained in Section 3.1.

Results: The results are reported in Table 3. The accuracies show that scne is also very effective in the sentence-level application. In this experiment, we observe that the larger $n_{\max}$ contributes to the performance improvement in sentence-level application by allowing our model to capture composed representations for longer phrases or sentences.

4 Conclusion

We proposed a simple yet effective unsupervised method to acquire general-purpose vector representations of words, phrases and sentences seamlessly, which is especially useful for languages whose word boundaries are not obvious, i.e., unsegmented languages. Although our method does not rely on any manually annotated resources or word segmenter, our extensive experiments show that our method outperforms the conventional approaches that depend on such resources.

Acknowledgments

We would like to thank anonymous reviewers for their helpful advice. This work was partially supported by JSPS KAKENHI grant 16H02789 to HS and 18J15053 to KF.

[4] Although Korean has spacing, word boundaries are not obviously determined by space.

[5] {food, song, music band name, manga, fictional character name, television series, drama, chemical compound, disease, taxon, city, island, country, year, business enterprise, public company, profession, university, language, book}

[6] For each language, we use 100MB of Wikipedia and SNS data as training corpora. For the SNS data, we use Sina Weibo for Chinese and Twitter for the rest.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5(1):135–146.

Daan van Esch. 2012. Leidon Weibo Corpus.

J.R. Firth. 1957. A synopsis of linguistic theory 1930-1955. *Studies in linguistic analysis*, pages 1–32.

Zellig S Harris. 1954. Distributional structure. *Word*, 10(2-3):146–162.

Benjamin Heinzerling and Michael Strube. 2018. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In *Proceedings of the 11th Language Resources and Evaluation Conference*, pages 2989–2993. European Language Resource Association.

Heewon Jeon. 2016. NIA (National Information Society Agency) Dictionary.

Peng Jin and Yunfang Wu. 2012. SemEval-2012 Task 4: Evaluating Chinese Word Similarity. In **SEM 2012: The First Joint Conference on Lexical and Computational Semantics – Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation (SemEval 2012)*, pages 374–377. Association for Computational Linguistics.

Sun Junyi. 2013. Jieba.

Geewook Kim, Kazuki Fukui, and Hidetoshi Shimodaira. 2018. Word-like character n-gram embedding. In *Proceedings of the 2018 EMNLP Workshop W-NUT: The 4th Workshop on Noisy User-generated Text*, pages 148–152. Association for Computational Linguistics.

Quoc Le and Tomas Mikolov. 2014. Distributed Representations of Sentences and Documents. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196. PMLR.

Mu Li, Jianfeng Gao, Changning Huang, and Jianfeng Li. 2003. Unsupervised Training for Overlapping Ambiguity Resolution in Chinese Word Segmentation. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing - Volume 17*, SIGHAN '03, pages 1–7. Association for Computational Linguistics.

Xiao Luo, Maosong Sun, and Benjamin K. Tsou. 2002. Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING '02, pages 1–7. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and their Compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

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. Association for Computational Linguistics.

Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. 2018. Unsupervised Learning of Sentence Embeddings Using Compositional n-Gram Features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Itsumi Saito, Kugatsu Sadamitsu, Hisako Asano, and Yoshihiro Matsuo. 2014. Morphological Analysis for Japanese Noisy Text based on Character-level and Word-level Normalization. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1773–1782. Dublin City University and Association for Computational Linguistics.

Yuya Sakaizawa and Mamoru Komachi. 2018. Construction of a Japanese Word Similarity Dataset. In *Proceedings of the 11th Language Resources and Evaluation Conference*, pages 948–951. European Language Resource Association.

Toshinori Sato. 2015. Neologism dictionary based on the language resources on the Web for Mecab.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725. Association for Computational Linguistics.

Yan Shao, Christian Hardmeier, and Joakim Nivre. 2018. Universal Word Segmentation: Implementation and Interpretation. *Transactions of the Association for Computational Linguistics*, 6:421–435.

Chengjie Sun, Chang-Ning Huang, Xiaolong Wang, and Mu Li. 2005. Detecting Segmentation Errors in Chinese Annotated Corpus. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 1–8. Association for Computational Linguistics.

Denny Vrandečić and Markus Krötzsch. 2014. Wikidata: A Free Collaborative Knowledge Base. *Communications of the ACM*, 57:78–85.

Shaonan Wang, Jiajun Zhang, and Chengqing Zong. 2017. Exploiting Word Internal Structures for Generic Chinese Sentence Representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 298–303. Association for Computational Linguistics.

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.

Jia Xu, Richard Zens, and Hermann Ney. 2004. Do We Need Chinese Word Segmentation for Statistical Machine Translation? In *ACL SIGHAN Workshop 2004*, pages 122–128. Association for Computational Linguistics.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural Word Segmentation with Rich Pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849. Association for Computational Linguistics.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level Convolutional Networks for Text Classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc.

Jinman Zhao, Sidharth Mudgal, and Yingyu Liang. 2018. Generalizing Word Embeddings using Bag of Subwords. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 601–606. Association for Computational Linguistics.

A Appendices

A.1 Experimental Details

A.1.1 Hyperparameters Tuning

For `skipgram`, we performed a grid search over $(h, \gamma) \in \{1, 5, 10\} \times \{0.01, 0.025\}$, where h is the size of context window and γ is the initial learning rate. For `sisg`, we performed a grid search over $(h, \gamma, n_{\min}, n_{\max}) \in \{1, 5, 10\} \times \{0.01, 0.025\} \times \{1, 3\} \times \{4, 8, 12\}$, where h is the size of context window, γ is the initial learning rate, n_{min} is the minimum length of character n-gram and n_{max} is the maximum length of character n-gram. For `pv-dbow`, `pv-dm` and `sent2vec`, we performed a grid search over $(h, \gamma) \in \{5, 10\} \times \{0.01, 0.05, 0.1, 0.2, 0.5\}$, where h is the size of context window and γ is the initial learning rate. For `sembei` and `scne`, we used the initial learning rate 0.01 and $n_{\min} = 1$. The maximum length of n-gram to consider $n_{\max}$ is grid searched over $\{4, 6, 8\}$ in the word and sentence similarity tasks. In the noun category prediction task, we used $n_{\max} = 8$ for `sembei` and the $n_{\max}$ of `scne` is grid searched over $\{4, 6, 8\}$. For sentiment analysis task, we tested both $n_{\max} = 8$ and $n_{\max} = 16$ for `sembei` and `scne` to see the effect of large $n_{\max}$. After carefully monitoring the loss curve and the performance in the word and sentence similarity tasks, we set the number of epochs 10 for all methods. In preliminary experiments, we also tested the number of epochs 20 for the word-segmentation-dependent baselines but there were no significant differences. In the two supervised downstream tasks, the learned vector representations are combined with the logistic regression classifier. The parameter C, which is the inverse of regularization strength of the classifier, is adjusted via a grid search over $C \in \{0.1, 0.5, 1, 5, 10\}$. Again, as explained in the main paper, the hyperparamters are grid searched on the determined validation set for all experiments.

A.1.2 Implementations

Here we provide the list of implementations of baselines which are used in our experiments. For `skipgram`[7], `sisg`[8], `sembei`[9], and `sent2vec`[10], we use the official implementations provided by the authors. Meanwhile, as for `pv-dbow` and `pv-dm`, we employ a widely-used implementation of Gensim library[11].

A.1.3 Word Segmenters and Word Dictionaries for Unsegmented Languages

Below we list the word segmentation tools and word dictionaries which are used in our experiments. We employed a widely-used word segmentation tool for each language.

For Chinese language, we used jieba[12] with its

[7] https://code.google.com/archive/p/word2vec/

[8] https://github.com/facebookresearch/fastText

[9] https://github.com/oshikiri/w2v-sembei

[10] https://github.com/epfml/sent2vec

[11] https://radimrehurek.com/gensim/models/doc2vec.html

[12] https://github.com/fxsjy/jieba

default dictionary[13] or with an extended dictionary[14], which fully supports both traditional and simplified Chinese characters.

For Japanese, we used MeCab[15] with its default dictionary called IPADIC[15] along with specially designed neologisms-extended dictionary called mecab-ipadic-NEologd[16]. Note that, because this extended dictionary *mecab-ipadic-NEologd* is specially designed to include many neologisms, there is a significant word coverage improvement by using this word dictionary as it can be seen in the Japanese noun category prediction task in the main paper.

For Korean, we used mecab-ko[17] with its default dictionary called mecab-ko-dic[18] along with another extended dictionary called NIADic[19].

A.1.4 Training Corpora

We prepared Wikipedia corpora and SNS corpora for Chinese, Japanese and Korean for our experiments. For the Wikipedia corpora, we used the first 10, 50, 100, 200 and 300MB of texts from the publicly available Wikipedia dumps[20]. The texts are extracted by using WikiExtractor tool[21]. For Chinese SNS corpus, we used 100MB of Leiden Weibo Corpus (van Esch, 2012) from the head. For Japanese and Korean SNS corpora, we collected Japanese and Korean tweets using Twitter Streaming API. We removed usernames and URLs from the SNS corpora. There were many informal words, emoticons and misspellings in the SNS corpora. We preserved them without preprocessing to see the effect of the noisiness of training corpora in our experiments.

A.1.5 Preprocess of Wikidata

For the noun category prediction task, we extracted noun words and their semantic categories from Wikidata (Vrandečić and Krötzsch, 2014) following Oshikiri (2017). We determined the semantic category set used in our experiments as follows: First, we collected Wikidata objects that have Chinese, Japanese, Korean and English labels. Next, we sorted the categories by the number of noun words, and removed categories (e.g., *Wikimedia category* or *Wikimedia template*) that do not represent any semantic category. We also removed out several categories that contain too many noun words (e.g., *human*) or too few noun words (e.g., *academic discipline*). Since there were several duplicated labels for different Wikidata objects, the number of nouns for each language is slightly different. Each category has at least 0.1k words and no more than 5k words. The numbers of extracted noun words that are used in our experiments were 22,468, 22,396 and 22,298 for Chinese, Japanese and Korean, respectively.

	skipgram$_{rich}$	sisg$_{rich}$	sembei	sembei-sum	scne
Wiki.	8.3	15.4	4.0	9.3	**24.1**
SNS	5.3	12.7	2.8	9.3	**23.0**
Diff.	-3.0	-2.7	-1.2	-0.0	-1.1

Table 4: Spearman rank correlations of the word similarity task on two different Japanese corpora.

A.1.6 Movie Review Datasets

In the main paper, three movie review datasets are used to evaluate the quality of sentence embeddings. We used 101,114, 55,837 and 200,000 movie reviews and their rating scores from Yahoo奇摩電影[22], Yahoo!映画[23] and Naver Movies[24] for Chinese, Japanese and Korean, respectively.

A.2 Additional Experiment on Japanese

In this section, we show the results of Japanese word similarity experiments. We use the datasets of Sakaizawa and Komachi (2018). It contains 4427 pairs of words with human similarity scores. We omit sentence similarity task since there is no public widely-used benchmark dataset for Japanese yet. Following the main paper, given a set of word pairs and their human annotated similarity scores, we calculated Spearman's rank correlation between the cosine similarities of the embeddings and the human scores. We use 2-fold

[13] https://github.com/fxsjy/jieba/blob/master/jieba/dict.txt
[14] https://github.com/fxsjy/jieba/blob/master/extra_dict/dict.txt.big
[15] http://taku910.github.io/mecab/
[16] https://github.com/neologd/mecab-ipadic-neologd
[17] https://bitbucket.org/eunjeon/mecab-ko
[18] https://bitbucket.org/eunjeon/mecab-ko-dic
[19] https://github.com/haven-jeon/NIADic
[20] https://dumps.wikimedia.org/
[21] https://github.com/attardi/wikiextractor
[22] https://github.com/fychao/ChineseMovieReviews
[23] https://github.com/dennybritz/sentiment-analysis/tree/master/data
[24] https://github.com/e9t/nsmc

cross validation for hyperparameters tuning. The same grid search is performed as explained in Section A.1.1. To see the effect of the noisiness of training corpora, we use two Japanese corpora, 100MB of Wikipedia corpus and 100MB of noisy SNS corpus (Twitter), which are also used in the Japanese noun category prediction task in the main paper. As seen in Table 4, the experiment results for Japanese are similar to those of Chinese in the main paper.