Subcharacter Information in Japanese Embeddings: When Is It Worth It?

Conference Paper · August 2018

DOI: 10.18653/v1/W18-2905

4 authors:

Marzena Karpinska
University of Massachusetts Amherst
7 PUBLICATIONS **7** CITATIONS
SEE PROFILE

Bofang Li
Renmin University of China
13 PUBLICATIONS **177** CITATIONS
SEE PROFILE

Anna Rogers
University of Copenhagen
22 PUBLICATIONS **830** CITATIONS
SEE PROFILE

Aleksandr Drozd
Tokyo Institute of Technology
26 PUBLICATIONS **428** CITATIONS
SEE PROFILE

Subcharacter Information in Japanese Embeddings: When Is It Worth It?

Marzena Karpinska[1], **Bofang Li**[2,3], **Anna Rogers**[4] **and Aleksandr Drozd**[3,5]
[1] Department of Language and Information Science, The University of Tokyo
[2] School of Information, Renmin University of China
[3] Department of Mathematical and Computing Science, Tokyo Institute of Technology
[4] Department of Computer Science, University of Massachusetts Lowell
[5] AIST- Tokyo Tech Real World Big-Data Computation Open Innovation Laboratory
karpinska@phiz.c.u-tokyo.ac.jp, libofang@ruc.edu.cn
arogers@cs.uml.edu, alex@blackbird.pw

Abstract

Languages with logographic writing systems present a difficulty for traditional character-level models. Leveraging the subcharacter information was recently shown to be beneficial for a number of intrinsic and extrinsic tasks in Chinese. We examine whether the same strategies could be applied for Japanese, and contribute a new analogy dataset for this language.

1 Introduction

No matter how big a corpus is, there will always be rare and out-of-vocabulary (OOV) words, and they pose a problem for the widely used word embedding models such as word2vec. A growing body of work on subword and character-level representations addresses this limitation in composing the representations for OOV words out of their parts (Kim et al., 2015; Zhang et al., 2015).

However, logographic writing systems consist of thousands of characters, varying in frequency in different domains. Fortunately, many Chinese characters (called *kanji* in Japanese) contain semantically meaningful components. For example, 木 (a standalone kanji for the word *tree*) also occurs as a component in 桜 (*sakura*) and 杉 (*Japanese cypress*).

We investigate the effect of explicit inclusion of kanjis and kanji components in the word embedding space on word similarity and word analogy tasks, as well as sentiment polarity classification. We show that the positive results reported for Chinese carry over to Japanese only partially, that the gains are not stable, and in many cases character ngrams perform better than character-level models. We also contribute a new large dataset for word analogies, the first one for this relatively low-resourced language, and a tokenizer-friendly version of its only similarity dataset.

2 Related Work

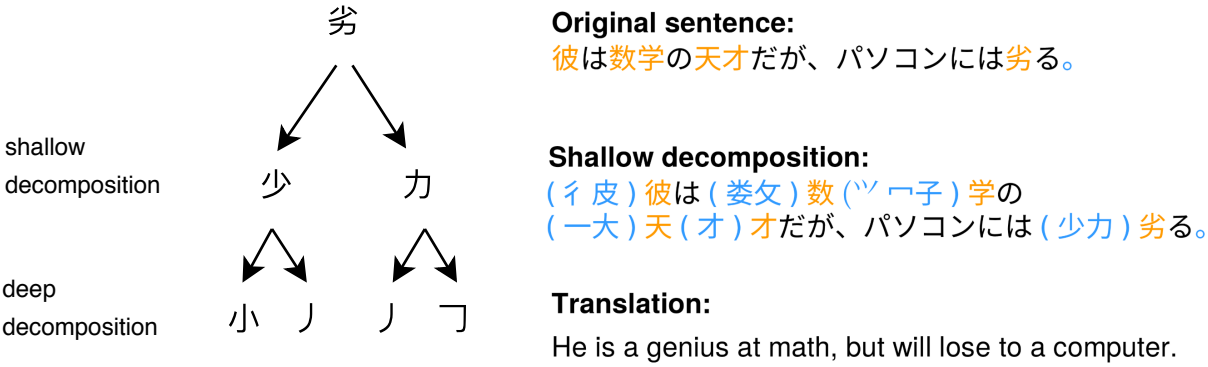
To date, most work on representing subcharacter information relies on language-specific resources that list character components[1]. A growing list of papers address various combinations of word-level, character-level and subcharacter-level embeddings in Chinese (Sun et al., 2014; Li et al., 2015; Yu et al., 2017). They have been successful on a range of tasks, including similarity and analogy (Yu et al., 2017; Yin et al., 2016), text classification (Li et al., 2015) sentiment polarity classification (Benajiba et al., 2017), segmentation, and POS-tagging (Shao et al., 2017).

Japanese kanjis were borrowed from Chinese, but it remains unclear whether these success stories could also carry over to Japanese. Chinese is an analytic language, but Japanese is agglutinative, which complicates tokenization. Also, in Japanese, words can be spelled either in kanji or in phonetic alphabets (*hiragana* and *katakana*), which further increases data sparsity. Numerous homonyms make this sparse data also noisy.

To the best of our knowledge, subcharacter information in Japanese has been addressed only by Nguyen et al. (2017) and Ke and Hagiwara (2017).

[1] Liu et al. (2017) showed the possibility of learning this information for any language through visual feature recognition.

Proceedings of the Workshop on the Relevance of Linguistic Structure in Neural Architectures for NLP, pages 28–37
Melbourne, Australia, July 19, 2018.



(a) Kanji decomposition levels

(b) Example sentence with shallow decomposition

Figure 1: Incorporating subcharacter information in Japanese

The former consider the language modeling task and compare several kinds of kanji decomposition, evaluating on model perplexity. Ke and Hagiwara (2017) propose to use subcharacter information instead of characters, showing that such a model performs on par with word and character-level models on sentiment classification, with considerably smaller vocabulary.

This study explores a model comparable to that proposed by Yu et al. (2017) for Chinese. We jointly learn a representation of words, kanjis, and kanjis' components, and we evaluate it on similarity, analogy, and sentiment classification tasks. We also contribute jBATS, the first analogy dataset for Japanese.

3 Incorporating Subcharacter Information

Kanji analysis depends on its complexity. Kanjis consisting of only 2-4 strokes may not be decomposable, or only containing 1-2 simple components (*bushu*). The more complex kanjis can usually be decomposed in analyzable bushu. This is referred to as shallow and deep decomposition (Figure 1a).

Nguyen et al. (2017) compared several decomposition databases in language modeling and concluded that shallow decomposition yields lower perplexity. This is rather to be expected, since many "atomic" bushu are not clearly meaningful. For example, Figure 1a shows the kanji 劣 ("to be inferior") as decomposable into 少 ("little, few") and 力 ("strength"). At the deep decomposition, only bushu 小 ("small") can be clearly related to the meaning of the original kanji 劣.

Hence, we use shallow decomposition. The bushu are obtained from IDS[2], a database that performed well for Nguyen et al. (2017). IDS is generated with character topic maps, which enables wider coverage[3] than crowd-sourced alternatives such as GlyphWiki.

In pre-processing each kanji was prepended the list of bushu (Figure 1b). Two corpora were used: the Japanese Wikipedia dump of April 01, 2018 and a collection of 1,859,640 Mainichi newspaper articles (Nichigai Associate, 1994-2009). We chose newspapers because this domain has a relatively higher rate of words spelled in kanji rather than hiragana.

As explained above, tokenization is not a trivial task in Japanese. The classic dictionary-based tokenizers such as MeCab or Juman, or their more recent ports such as Kuromoji do not handle OOV very well, and the newer ML-based tokenizers such as TinySegmenter or Micter are also not fully reliable. We tokenized the corpora with MeCab using a weekly updated neologism dictionary[4], which yielded roughly 357 million tokens for Mainichi and 579 for Wiki[5]. The tokenization was highly inconsistent: for example, 満腹感 ("feeling full") is split into 満腹 ("full stomach") and 感 ("feeling"), but 恐怖感 ("feeling fear") is a single word, rather than 恐怖 + 感 ("fear" and "feeling"). We additionally pre-processed the corpora to correct the tokenization for all the affixes

[2] http://github.com/cjkvi/cjkvi-ids

[3] A limitation of IDS is that it does not unify the representations of several frequent bushu, which could decrease the overall quality of the resulting space (e.g. 心 "heart" is being pictured as 心, 忄 and 小 depending on its position in kanji).

[4] http://github.com/neologd/mecab-ipadic-neologd

[5] The Wikipedia tokenized corpus is available at http://vecto.space/data/corpora/ja

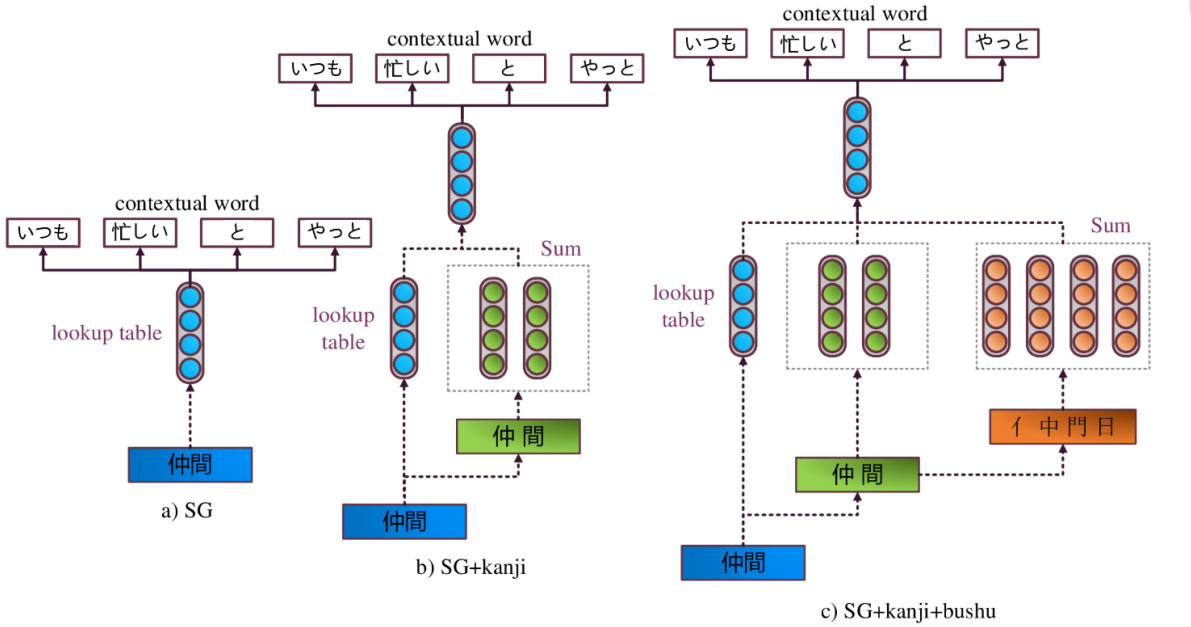


Figure 2: Model architecture of SG, SG+kanji, and SG+kanji+bushu. Example sentence: いつも 忙しい 仲間 と やっと 会え た ("I have finally met with my busy colleague."), window size 2.

in jBATS (section 5).

4 Model architecture

4.1 Baselines

Original SG. Skip-Gram (SG) (Mikolov et al., 2013) is a popular word-level model. Given a target word in the corpus, SG model uses the vector of this target word to predict its contextual words.

FastText. FastText (Bojanowski et al., 2017) is a state-of-the-art subword-level model that learns morphology from character n-grams. In this model, each word is considered as the sum of all the character n-grams.

4.2 Characters and subcharacters

Characters (*kanji*). To take individual kanji into account we modified SG by summing the target word vector w with vectors of its constituent characters c_1, and c_2. This can be regarded as a special case of FastText, where the minimal n-gram size and maximum n-gram size are both set to 1. Our model is similar to the one suggested by Yu et al. (2017), who learn Chinese word embeddings based on characters and sub-characters. We refer to this model as SG+kanji.

Subcharacters (*bushu*). Similarly to characters, we sum the vector of the target word, its constituent characters, and their constituent bushu to incorporate the bushu information. For example, Figure 3 shows that the vector of the word 仲間, the vectors of characters 仲 and 間, and the vectors of bushu イ, 中, 門, 日 are summed to predict the contextual words. We refer to this model as SG+kanji+bushu.

Expanding vocabulary. FastText, SG+kanji and SG+kanji+bushu models can be used to compute the representation for any word as a sum of the vectors of its constituents. We collect the vocabulary of all the datasets used in this paper, calculate the vectors for any words missing in the embedding vocabulary, and add them. Such models will be referred to as MODEL+OOV.

4.3 Implementation

All models were implemented in Chainer framework (Tokui et al., 2015) with the following parameters: vector size 300, batch size 1000, negative sampling size 5, window size 2. For performance reasons all models were trained for 1 epoch. Words, kanjis and bushu appearing less than 50 times in the corpus were ignored. The optimization function was Adam (Kingma and Ba, 2014). The n-gram size of FastText[6] is set to 1, for

[6]The original FastText code[7] has some inherent differences from our Chainer implementation, as it was designed for CPU only. On each CPU thread, it directly updates the weight parameters after evaluation of each sample. To take the advantage of GPU, we use mini-batch (size 1000) to par-

	Relation	Example		Relation	Example
Inflections	I01 Verb: u-form >a-form	使う: 使わ	Lexicography	L01 hypernyms (animals)	カメ: 爬虫/脊椎動物/
	I02 Verb: u-form >o-form	受ける: 受けよ		L02 hypernyms (misc.)	椅子: 支え/器具/道具/人工物...
	I03 Verb: u-form >e-form	起きる: 起きれ		L03 hyponyms (misc.)	肉: 牛肉/牛/ビーフ/鳥肉/...
	I04 Verb: u-form >te-form	会う : 会っ		L04 meronyms (substance)	バッグ: 革/生地/布/プラスチック
	I05 Verb: a-form >o-form	書か : 書こ		L05 meronyms (member)	鳥: 群れ/家畜
	I06 Verb: o-form >e-form	歌お: 歌え		L06 meronyms (part)	アカデミア: 大学/大学院/学院...
	I07 Verb: e-form >te-form	勝て: 勝っ		L07 synonyms (intensity)	つまらない,退屈/くだらない/...
	I08 i-Adj.: i-form >ku-form	良い: 良く		L08 synonyms (exact)	赤ちゃん: 赤ん坊/ベビー
	I09 i-Adj: i-form >ta-form	良い: 良かっ		L09 antonyms (gradable)	大きい: 小さい/ちび/ちっちゃい...
	I10 i-Adj.: ku-form >ta-form	良く: 良かっ		L10 antonyms (binary)	出口: 入り口/入口
Derivation	D01 na-adj + ”化”	活性: 活性化	Encyclopedia	E01 capital: country	ロンドン: イギリス/英国
	D02 i-adj + ”さ”	良い: 良さ		E02 country: language	フランス : フランス語
	D03 noun + ”者”	消費: 消費者		E03 jp. prefecture: city	沖縄県: 那覇/那覇市
	D04 ”不” + noun	人気: 不人気		E04 name: nationality	アリストテレス: ギリシャ人
	D05 noun + ”会”	運動: 運動会		E05 name: occupation	アリストテレス: 哲学者
	D06 noun/na-adj. + ”感”	存在: 存在感		E06 onomatopoeia : feeling	ドキドキ: 緊張/恐怖
	D07 noun/na-adj. + ”性”	可能: 可能性		E07 company: product	日産: 車/自動車
	D08 noun/na-adj. + ”力”	影響: 影響力		E08 object: usage	ギター : 弾く
	D09 ”大”+ noun/na-adj.	好き: 大好き		E09 polite terms	おっしゃる: 申し上げる
	D10: (in)transitive verb	起きる:起こす		E10 object: color	カラス: 黒/黒い

Table 1: jBATS: structure and examples

reliable comparison with our character model. We experimented with 1/2 of Mainichi corpus while developing the models, and then trained them on full Mainichi and Wikipedia. All sets of embeddings are available for download[8].

For SG+kanji+bushu model there were 2510 bushu in total, 1.47% of which were ignored in the model since they were not in the standard UTF-8 word (“w) encoding. This affected 1.37% of tokens in Wikipedia.

5 Evaluation: jBATS

We present jBATS[9], a new analogy dataset for Japanese that is comparable to BATS (Gladkova et al., 2016), currently the largest analogy dataset for English. Like BATS, jBATS covers 40 linguistic relations which are listed in Table 1. There are 4 types of relations: inflectional and derivational morphology, and encyclopedic and lexicographic semantics. Each type has 10 categories, with 50 word pairs per category (except for E03 which has 47 pairs, since there are only 47 prefectures). This enables generation of 97,712 analogy questions.

The inflectional morphology set is based on the traditional Japanese grammar (Teramura, 1982) which lists 7 different forms of *godan*, *shimoichidan* and *kamiichidan* verbs, as well as 5 forms of *i*-adjectives. Including the past tense form, there are 8 and 6 forms for verbs and adjectives respectively. All categories were adjusted to the MeCab tokenization. After excluding redundant or rare forms there were 5 distinctive forms for verbs and 3 for adjectives, which were paired to form 7 verb and 3 adjective categories.

The derivational morphology set includes 9 highly productive affixes which are usually represented by a single kanji character, and a set of pairs of transitive and intransitive verbs which are formed with several infix patterns.

The encyclopedic and lexicographic semantics sections were designed similarly to BATS (Gladkova et al., 2016), but adjusted for Japanese. For example, UK counties were replaced with Japanese prefectures. The E09 *animal-young* category of BATS would be rendered with a prefix in Japanese, and was replaced with plain: honorific word pairs, a concept highly relevant for the Japanese culture.

All tokens were chosen based on their frequencies in BCCWJ[10] (Maekawa, 2008), the Balanced Corpus of Contemporary Written Japanese, and the Mainichi newspaper corpus described in Section 3. We aimed to choose relatively frequent and not genre-specific words. For broader categories (adjectives and verbs) we balanced between BCCWJ and Mainichi corpora, choosing items of mean frequencies between 3,000 and 100,000

allelize training.

[8] http://vecto.space/data/embeddings/ja

[9] http://vecto.space/projects/jBATS

[10] http://pj.ninjal.ac.jp/corpus_center/bccwj/en/freq-list.html

whenever possible.

6 Results

6.1 Word similarity

The recent Japanese word similarity dataset (Sakaizawa and Komachi, 2017) contains 4,851 word pairs that were annotated by crowd workers with agreement 0.56-0.69. Like MEN (Bruni et al., 2014) and SimLex (Hill et al., 2015), this dataset is split by parts of speech: verbs, nouns, adjectives and adverbs. We refer to this dataset as jSIM.

The division by parts of speech is relevant for this study: many Japanese adverbs are written mostly in hiragana and would not benefit from bushu information. However, some pairs in jSIM were misclassified. Furthermore, since this dataset was based on paraphrases, many pairs contained phrases rather than words, and/or words in forms that would not be preserved in a corpus tokenized the Mecab style (which is the most frequently used in Japanese NLP). Therefore, for embeddings with standard pre-processing jSIM would have a very high OOV rate. The authors of jSIM do not actually present any experiments with word embeddings.

We have prepared 3 versions of jSIM that are summarized in Table 2. The *full* version contains most word pairs of the original dataset (except those which categories were ambiguous or mixed), with corrected POS attribution in 2-5% of pairs in each category[11]: for example, the pair 苛立たしい - 忌ま忌ましい was moved from verbs to adjectives. The *tokenized* version contains only the items that could be identified by a Mecab-style tokenizer, and had no more than one content-word stem: e.g. this would exclude phrases like 早く来る. However, many of the remaining items could become ambiguous when tokenized: 終わった would become 終わっ た – and 終わっ could map to 終わった, 終わって, 終わっちゃう, etc., and therefore be more difficult to detect in the similarity task. Thus we also prepared the *unambiguous* subset which contains only the words that could still be identified unambiguously even when tokenized (for example, 迷う remains 迷う). All these versions of jSIM are available for download[12].

Table 3 shows the results on all 3 datasets on all models, trained on the full Mainichi corpus, a half Mainichi corpus, and Wikipedia. The strongest effect for inclusion of bushu is observed in the OOV condition: in all datasets the Spearman's correlations are higher for SG+kanji+bushu than for other SG models, which suggests that this information is indeed meaningful and helpful. This even holds for the *full* version, where up to 90% vocabulary is missing and has to be composed. For in-vocabulary condition this effect is noticeably absent in Wikipedia (perhaps due to the higher ratio of names, where the kanji meanings are often irrelevant).

Version	Adj.	Adv.	Nouns	Verbs	Total
Original	960	902	1103	1464	4429
Full	879	893	1104	1507	4383
Tokenized	642	774	947	427	2790
Unambiguous	448	465	912	172	1997

Table 2: The size of the original and modified Japanese similarity datasets (in word pairs)

However, in most cases the improvement due to inclusion of bushu, even when it is observed, is not sufficient to catch up with the FastText algorithm, and in most cases FastText has substantial advantage. This is significant, as it might warrant the review of the previous results for Chinese on this task: of all the studies on subcharacter information in Chinese that we reviewed, only one explicitly compared their model to FastText (Benajiba et al., 2017), and their task was different (sentiment analysis).

In terms of parts of speech, the only clear effect is for the adjectives, which we attribute to the fact that many Japanese adjectives contain a single kanji character, directly related to the meaning of the word (e.g. 惜しい). The adjectives category contains 55.45% such words, compared to 14.78% for nouns and 23.71% for adverbs in the *full* jSIM (the ratio is similar for *Tokenized* and *Unambiguous* sets). On the other hand, all jSIM versions have over 70% of nouns with more than one kanji; some of them may not be directly related to the meaning of the word, and increase the noise. Ac-

[11] Division between adjectives and adverbs is problematic for the Japanese adverbial forms of adjectives, such as 安い → 安く. There were 228 such pairs in total. Since we focus on the kanji, we grouped them with the adjectives, as in the original dataset.

[12] http://vecto.space/projects/jSIM

	Model	Full				Tokenized				Unambiguous			
		adj	adv	noun	verb	adj	adv	noun	verb	adj	adv	noun	verb
Mainichi 1/2	FastText	**.366**	.190	**.331**	.355	**.392**	.285	**.333**	**.381**	.377	.232	**.328**	.337
	SG	.321	**.346**	.274	.311	.352	**.364**	.280	.341	.340	**.362**	.274	.304
	SG+kanji	.339	.290	.280	.294	.371	.330	.285	.345	.369	.305	.279	.302
	SG+kanji+bushu	.355	.300	.276	**.391**	.380	.356	.279	.375	**.384**	.326	.274	**.393**
	OOV rate per category	.659	.616	.328	.934	.506	.295	.232	.372	.462	.318	.235	.436
	FastText+OOV	**.435**	.153	**.213**	**.241**	**.416**	.185	**.259**	**.359**	**.434**	.124	**.252**	**.373**
	SG+kanji+OOV	.344	.195	.152	.210	.279	.235	.192	.307	.309	.211	.179	.327
	SG+kanji+bushu+OOV	.329	**.220**	.146	.230	.272	**.261**	.188	.318	.311	**.242**	.177	.372
Mainichi	FastText	.399	.277	**.336**	**.345**	**.436**	.296	**.337**	.355	.397	.310	**.328**	**.345**
	SG	.345	**.336**	.280	.246	.362	.333	.282	.295	.367	**.359**	.274	.246
	SG+kanji	.366	.321	.269	.334	.391	**.354**	.272	**.363**	.399	.348	.262	.334
	SG+kanji+bushu	**.405**	.318	.288	.315	.427	.311	.291	.353	**.444**	.341	.282	.315
	OOV rate per category	.582	.586	.272	.922	.389	.260	.164	.262	.384	.288	.166	.320
	FastText+OOV	**.448**	.184	**.245**	**.242**	**.438**	.222	**.286**	**.410**	**.453**	.202	**.275**	**.405**
	SG+kanji+OOV	.323	**.195**	.175	.210	.293	**.262**	.210	.353	.341	**.250**	.197	.363
	SG+kanji+bushu+OOV	.348	.171	.178	.201	.318	.231	.223	.330	.373	.249	.210	.315
Wikipedia	FastText	**.405**	.296	**.333**	**.341**	**.440**	.298	**.334**	.348	**.402**	.330	**.325**	**.341**
	SG	.309	**.298**	.299	.320	.312	.315	.299	**.382**	.307	**.345**	.296	.320
	SG+kanji	.334	**.298**	.270	.326	.331	**.327**	.275	.380	.324	.334	.271	.326
	SG+kanji+bushu	.321	.285	.282	.270	.312	.295	.287	.364	.326	.315	.279	.270
	OOV rate per category	.578	.591	.225	.909	.393	.269	.112	.192	.384	.301	.112	.203
	FastText+OOV	**.451**	**.186**	**.242**	**.243**	**.442**	.225	**.281**	**.400**	**.455**	.219	**.270**	**.402**
	SG+kanji+OOV	.296	.179	.146	.185	.240	**.240**	.191	.325	.270	.239	.184	.278
	SG+kanji+bushu+OOV	.313	.183	.159	.171	.249	.238	.208	.315	.292	**.254**	.197	.243

Table 3: Spearman's correlation with human similarity judgements. Boldface indicates the highest result on a given corpus (separately for in-vocabulary and OOV conditions). Shaded numbers indicate the highest result among the three Skip-Gram models.

cordingly, we observe the weakest effect for inclusion of bushu. However, the ratio of 1-kanji words for verbs is roughly the same as for the adjectives, but the pattern is less clear.

Adverbs are the only category in which SG clearly outperforms FastText. This could be due to a high proportion of hiragana (about 50% in all datasets), which as single-character ngrams could not yield very meaningful representations. Also, the particles と and に, important for adverbs, are lost in tokenization.

6.2 jBATS

In this paper, we consider two methods for the word analogy task. **3CosAdd** (Mikolov et al., 2013) is the original method based on linear offset between 2 vector pairs. Given an analogy a:a' :: b:b' (a is to a' as b is to b'), the answer is calculated as $b' = argmax_{d \in V}(cos(b', b - a + a'))$, where $cos(u, v) = \frac{u \cdot v}{||u|| \cdot ||v||}$

LRCos (Drozd et al., 2016) is a more recent and currently the best-performing method. It is based on a set of word pairs that have the same relation. For example, given a set of pairs such as *husband:wife*, *uncle:aunt*, all right-hand words are considered to be exemplars of a class ("women"), and logistic regression classifier is trained for that class. The answer (e.g. *queen*) is determined as the word vector that is the most similar to the source word (e.g. *king*), but is likely to be a *woman*:

$$b' = argmax_{b' \in V}(P_{(b' \in class)} * cos(b', b))$$

Figure 3 shows that the overall pattern of accuracy for jBATS is comparable to what Gladkova et al. (2016) report for English: derivational and inflectional morphology are much easier than either kind of semantics. In line with the results by Drozd et al. (2016), LRCos significantly outperforms 3CosAdd, achieving much better accuracy on some encyclopedic categories with which

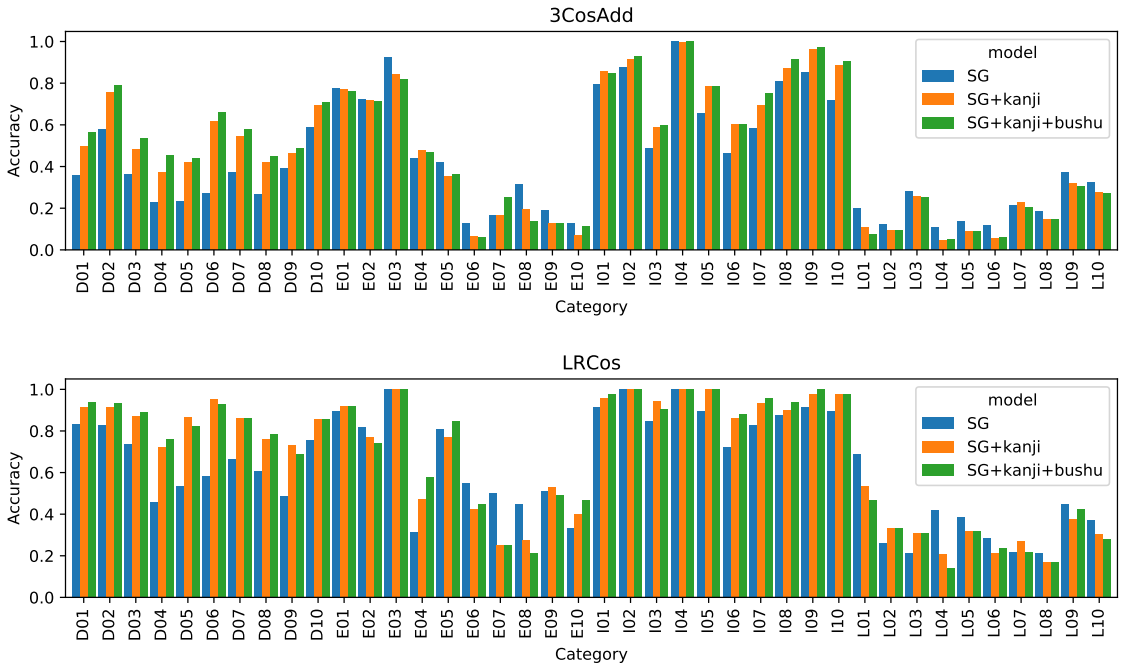


Figure 3: Accuracy on jBATS with 3CosAdd and LRCos methods (see Table 1 for the codes on x-axis).

3CosAdd does not cope at all. Lexicographic semantics is a problem, as in English, because synonyms or antonyms of different words do not constitute a coherent semantic class by themselves.

	Model	inf.	der.	enc.	lex.
Mainichi 1/2	FastText	.902	.770	.237	.075
	SG	.785	.452	**.318**	**.110**
	SG+kanji	.892	.771	.314	.102
	SG+kanji+bushu	**.912**	**.797**	.253	.083
	OOV rate per category	.070	.076	.408	.256
	FastText+OOV	.846	.758	.146	.090
	SG+kanji+OOV	.856	.747	**.181**	**.102**
	SG+kanji+bushu+OOV	**.883**	**.768**	.163	.088
Mainichi	FastText	.883	.648	.232	.093
	SG	.853	.496	**.370**	**.133**
	SG+kanji	.912	.676	.330	.123
	SG+kanji+bushu	**.926**	**.710**	.318	.118
	OOV rate per category	.022	.056	.346	.204
	FastText+OOV	.861	.746	.173	.114
	SG+kanji+OOV	**.912**	.676	**.330**	**.123**
	SG+kanji+bushu+OOV	.893	**.705**	.215	.094
Wikipedia	FastText	**.881**	.663	.242	.088
	SG	.743	.457	**.484**	**.170**
	SG+kanji	.834	.638	.422	.112
	SG+kanji+bushu	.851	**.694**	.425	.100
	OOV rate per category	.036	.060	.322	.142
	FastText+OOV	**.846**	**.750**	.158	**.127**
	SG+kanji+OOV	.794	.639	**.297**	.098
	SG+kanji+bushu+OOV	.833	.671	.293	.102

Table 4: Word analogy task accuracy (LRCos). Boldface indicates the highest result for a corpus, and the shaded numbers indicate the highest result among three Skip-Gram models.

Table 4 shows the average results per relation type for the better-performing LRCos (the pattern of results was similar for 3CosAdd). The morphology categories behave similarly to adjectives in the similarity task: the SG+kanji beats the original SG by a large margin on inflectional and derivational morphology categories, and bushu improve accuracy even further. In this task, these models also win over FastText. However, these are the categories in which the words either contain a single kanji, or (in derivational morphology) a single kanji affix needs to be identified. Semantic categories contain a variety of nouns, mostly consisting of several kanjis with various morphological patterns. Moreover, many proper nouns as well as animal species are written in katakana, with no kanjis at all. This could be the reason why information from kanjis and bushu are not helpful or even detrimental in the semantic questions.

There is a clear corpus effect in that the encyclopedic semantic questions are (predictably) more successful with Wikipedia than with Mainichi, but at the expense of morphology. This could be interpreted as confirmation of the dependence of the current analogy methods on similarity (Rogers et al., 2017): all words cannot be close to all other words, so a higher ratio of some relation type has

Error type	Example	Predicted	Percentage
correct stem, wrong form	買う : 買え :: 借りる : [借りれ]	借り	28.0%
same semantic category	アメリカ : 英語 :: イラン : [ペルシア語]	トルコ語	25.0%
antonym, correct form	深い : 深さ :: 低い : [低さ]	高さ	10.0%
antonym, wrong form	面白い : 面白さ :: 高い : [高さ]	低い	3.0%
related to target pair	アンドラ : カタルーニャ語 :: アメリカ : [英語]	米国	8.5%
wrong stem, correct form	持つ : 持て :: 借りる : [借りれ]	買え	5.5%
duplicated token	もらう : あげる :: 内（うち） : [外]	うち	5.0%
synonym, correct form	悪い : 悪さ :: すごい : [すごさ]	器用さ	1.0%
synonym, wrong form	ほしい : ほしさ :: 固い : [固さ]	堅い	1.5%
orthography related	減る : 増える :: オン : [オフ]	フォー	1.0%
related to the source pair	前 : 次 :: 内 : [外]	下記	0.5%
alternative spelling	イスラエル : ヘブライ語 :: イラン : [ペルシア語]	ペルシャ語	0.5%
unrelated	痛い : 痛さ :: 大きい : [大きさ]	仮種皮	10.5%

Table 5: jBATS: error analysis.

to come with a decrease in some other.

6.3 Sentiment analysis

The binary sentiment classification accuracy was tested with the Rakuten reviews dataset by Zhang and LeCun (2017). Although Benajiba et al. (2017) report that incorporating subcharacter information provided a boost in accuracy on this task in Chinese, we did not confirm this to be the case for Japanese. Table 6 [13] shows that the accuracy for all models ranged between 0.92-0.93 (consistent with the results of Zhang and LeCun (2017)), so no model had a clear advantage.

Model	Main.1/2	Mainichi	Wiki
FastText	.919	.921	.920
SG	**.921**	.920	**.921**
SG+kanji	**.921**	**.924**	.919
SG+kanji+bushu	.918	.920	**.921**
OOV rate per category	.220	.220	.212
FastText+OOV	.926	.927	**.922**
SG+kanji+OOV	**.929**	**.930**	**.922**
SG+kanji+bushu+OOV	.925	.927	**.922**

Table 6: Sentiment analysis accuracy

The lack of positive effect for inclusion of kanji and bushu is to be expected, as we found that most of the dataset is written informally, in hiragana, even for words that are normally written with kanjis. Once again, this shows that the results of incorporating (sub)character information in Japanese are not the same as in Chinese, and depend on the task and domain of the texts.

Interestingly, the accuracy is just as high for all OOV models, even though about 20% of the vocabulary had to be constructed.

[13] The Chainer framework (Tokui et al., 2015) is used to implement the CNN classifier with default settings.

7 Discussion

7.1 Error analysis

We conducted manual analysis of 200 mispredictions of 3CosAdd method in I03, D02, E02 and L10 categories (50 examples in each). The percentage of different types of errors is shown in Table 5. Overall, most mistakes are interpretable, and only 10.5% of mispredicted vectors are not clearly related to the source words.

The most frequent example of mis-classification was predicting the wrong form but with the correct stem, especially in morphological categories. This is consistent with what Drozd et al. (2016) report for English and was especially frequent in the I03 and D02 categories (76% and 36% of errors per category respectively). It is not surprising since these categories consist of verbs (I03) and adjectives (D02). Furthermore, in 25% of cases the assigned item was from the same semantic category (for example, colours) and in 13% of case an antonym was predicted. Other, though relatively less frequent mistakes include semantic relations like predicting synonyms of the given word, words (or single kanji) related to either target or source pair, or simply returning the same token. Words which were not related in any way to any source word were very rare.

7.2 Vector neighborhoods

Table 7 shows that the shared semantic space of words, kanjis and bushu is indeed shared. For example, the bushu 疒 (*yamaidare* “the roof from illness”) is often used in kanjis which are related to a disease. Therefore kanji like 症 (“disease”) would,

疒 *yamaidare* (the roof from illness)	豸 *najina-hen* (devine beast, insect without legs)
患(sickness) 症(disease) 妊 (pregnancy)	爭(to fight, to compete) 蝶(butterfly)
臓 (internal organs, bowels) 腫 (tumor)	皃(shape) 貌(shape, silhouette) 豹(leopard)
インフルエザ (influenza)	獅子 (lion, king of beasts)
関節リウマチ (articular rheumatism)	同流 (same origin, same school)
リューマチ (rheumatism) リウマチ(rheumatism)	本性(true nature, human nature)
メタボリックシンドローム (metabolic syndrome)	弥勒(Maitreya Buddha) 無頼 (villain, scoundrel)

Table 7: Example bushu: closest single kanji (upper row) and multiple kanji/katakana (lower row) for SG+kanji+bushu model.

of course, be similar to 疒 in the vector space. Interestingly, we also find that its close neighbors include kanjis that do not have this bushu, but are related to disease, such as 腫 and 患. Furthermore, even words written only in katakana, like インフルエザ, are correctly positioned in the same space. Similar observations can be made for bushu 豸(*mujina-hen*) which represents a divine beast, insects without legs, animals with long spine, or a legendary Chinese beast *Xiezhi*.

7.3 Stability of the similarity results

Our similarity experiments showed that in many cases the gain of any one model over the other is not very significant and would not be reproduced in a different run and/or a different corpus. This could be due to skewed frequency distribution or the general instability of embeddings for rare words, recently demonstrated for word2vec (Wendlandt et al., 2018).

One puzzling observation is that sometimes the smaller corpus yielded better embeddings. Intuitively, the larger the corpus, the more informative distributional representations can be obtained. However, Table 3 shows that for adverbs and verbs the *full* and *tokenized* versions of jSIM a half of Mainichi was actually significantly better than the full Mainichi. It is not clear whether it is due to a lucky random initialization or some other factors.

8 Conclusion

This study presented the first evaluation of subcharacter-level distributional representations of Japanese on similarity, analogy and sentiment classification tasks. We show that the success of this approach in Chinese is transferable to Japanese only partly, but it does improve the performance of Skip-Gram model in kanji-rich domains and for tasks relying on mostly single-kanji vocabulary or morphological patterns. The effect may be stronger with a better sent of model hyperparameters, which we have not explored here, or in some other task. However, in our experiments we found that even enhanced Skip-Gram was consistently inferior to single-character ngram FastText, which has not been used as a baseline in most work on Chinese subcharacter-level embeddings.

We also contribute jBATS, the first analogy dataset for this relatively low-resourced language, and a revision of its only similarity dataset that can now be used with standard tokenized corpora. All models, datasets and embeddings are available in the `Vecto`[14] library.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant No. JP17K12739, JST CREST Grant No. JPMJCR1687 and National Natural Science Foundation of China Grant No.61472428.

References

Yassine Benajiba, Or Biran, Zhiliang Weng, Yong Zhang, and Jin Sun. 2017. The Sentimental Value of Chinese Sub-Character Components. In *Proceedings of the 9th SIGHAN Workshop on Chinese Language Processing*, pages 21–29, Taipei, Taiwan, December 1, 2017. Association for Computational Linguistics.

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

Elia Bruni, Nam-Khanh Tran, and Marco Baroni. 2014. Multimodal distributional semantics. *J. Artif. Intell. Res.(JAIR)*, 49(2014):1–47.

Aleksandr Drozd, Anna Gladkova, and Satoshi Matsuoka. 2016. Word embeddings, analogies, and machine learning: Beyond king - man + woman = queen. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3519–3530, Osaka, Japan, December 11-17.

[14] `http://vecto.space`

Anna Gladkova, Aleksandr Drozd, and Satoshi Matsuoka. 2016. Analogy-based detection of morphological and semantic relations with word embeddings: What works and what doesn't. In *Proceedings of the NAACL-HLT SRW*, pages 47–54, San Diego, California, June 12-17, 2016. ACL.

Felix Hill, Roi Reichart, and Anna Korhonen. 2015. Simlex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*, 41(4):665–695.

Yuanzhi Ke and Masafumi Hagiwara. 2017. Radical-level Ideograph Encoder for RNN-based Sentiment Analysis of Chinese and Japanese. *arXiv:1708.03312 [cs]*.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2015. Character-aware neural language models. *arXiv preprint arXiv:1508.06615*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Yanran Li, Wenjie Li, Fei Sun, and Sujian Li. 2015. Component-enhanced Chinese character embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 829–834, Lisbon, Portugal, 17-21 September 2015. Association for Computational Linguistics.

Frederick Liu, Han Lu, Chieh Lo, and Graham Neubig. 2017. Learning Character-level Compositionality with Visual Features. pages 2059–2068. Association for Computational Linguistics.

Kikuo Maekawa. 2008. Compilation of the Balanced Corpus of Contemporary Written Japanese in the KOTONOHA Initiative. In *Universal Communication, 2008. ISUC'08. Second International Symposium On*, pages 169–172. IEEE.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119.

Viet Nguyen, Julian Brooke, and Timothy Baldwin. 2017. Sub-character Neural Language Modelling in Japanese. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 148–153.

Nichigai Associate. 1994-2009. CD-Mainichi Shimbun de-ta shu (1994-2009).

Anna Rogers, Aleksandr Drozd, and Bofang Li. 2017. The (Too Many) Problems of Analogical Reasoning with Word Vectors. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 135–148.

Yuya Sakaizawa and Mamoru Komachi. 2017. Construction of a Japanese Word Similarity Dataset. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Yan Shao, Christian Hardmeier, Jorg Tiedemann, and Joakim Nivre. 2017. Character-based Joint Segmentation and POS Tagging for Chinese using Bidirectional RNN-CRF. page 11.

Yaming Sun, Lei Lin, Nan Yang, Zhenzhou Ji, and Xiaolong Wang. 2014. Radical-Enhanced Chinese Character Embedding. In *Neural Information Processing*, Lecture Notes in Computer Science, pages 279–286. Springer, Cham.

Hideo Teramura. 1982. *Nihongo no shintakusu to imi (Japanese syntax and meaning)*. Kuroshio Shuppan.

Seiya Tokui, Kenta Oono, Shohei Hido, and Justin Clayton. 2015. Chainer: a next-generation open source framework for deep learning. In *Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS)*, volume 5.

Laura Wendlandt, Jonathan K. Kummerfeld, and Rada Mihalcea. 2018. Factors Influencing the Surprising Instability of Word Embeddings. *arXiv:1804.09692 [cs]*.

Rongchao Yin, Quan Wang, Peng Li, Rui Li, and Bin Wang. 2016. Multi-Granularity Chinese Word Embedding. pages 981–986. Association for Computational Linguistics.

Jinxing Yu, Xun Jian, Hao Xin, and Yangqiu Song. 2017. Joint Embeddings of Chinese Words, Characters, and Fine-grained Subcharacter Components. pages 286–291. Association for Computational Linguistics.

Xiang Zhang and Yann LeCun. 2017. Which Encoding is the Best for Text Classification in Chinese, English, Japanese and Korean? *arXiv preprint arXiv:1708.02657*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *NIPS*, pages 649–657.