A Character-based Approach to Distributional Semantic Models: Exploiting Kanji Characters for Constructing Japanese Word Vectors

Akira Utsumi

Department of Informatics
The University of Electro-Communications
1-5-1 Chofugaoka, Chofushi, Tokyo 182-8585, Japan
utsumi@uec.ac.jp

Abstract

Many Japanese words are made of kanji characters, which themselves represent meanings. However traditional word-based distributional semantic models (DSMs) do not benefit from the useful semantic information of kanji characters. In this paper, we propose a method for exploiting the semantic information of kanji characters for constructing Japanese word vectors in DSMs. In the proposed method, the semantic representations of kanji characters (i.e, kanji vectors) are constructed first using the techniques of DSMs, and then word vectors are computed by combining the vectors of constituent kanji characters using vector composition methods. The evaluation experiment using a synonym identification task demonstrates that the kanji-based DSM achieves the best performance when a kanji-kanji matrix is weighted by positive pointwise mutual information and word vectors are composed by weighted multiplication. Comparison between kanji-based DSMs and word-based DSMs reveals that our kanji-based DSMs generally outperform latent semantic analysis, and also surpasses the best score word-based DSM for infrequent words comprising only frequent kanji characters. These findings clearly indicate that kanji-based DSMs are beneficial in improvement of quality of Japanese word vectors.

1. Introduction

Distributional semantic models (henceforth, DSMs) or semantic space models are models for semantic representations and for the way where semantic representations are built (Turney and Pantel, 2010). The semantic content is represented by a high-dimensional vector, and these vectors are constructed from large bodies of text (i.e., corpus) by observing distributional statistics of word occurrence. Despite their simplicity, DSMs have proved highly useful for a variety of NLP applications that involve measuring the similarity of meaning between words, phrases, and documents, such as information retrieval, document clustering, collaborative filtering, and essay scoring (Landauer et al., 2007; Widdows, 2004). Furthermore, DSMs have also provided a useful framework for cognitive modeling, for example, similarity judgment, semantic priming, and metaphor comprehension (Jones et al., 2006; Landauer and Dumais, 1997; Utsumi, 2011).

DSMs generally assume that a word is a basic unit of semantic representation. Words cannot be decomposed of smaller semantic units (except for some morphemes such as prefix and suffix). For example, English words “select” and “erect” share a sequence of characters ”ect,” but this does not imply that these two words are semantically related. This assumption is essentially appropriate for many languages using phonograms or phonetic-based symbols such as alphabets. In contrast, some languages such as Japanese and Chinese use logograms, namely symbols which represent meanings. The logogram used in the Japanese language is called *kanji characters (or kanji)* (i.e., Chinese characters adopted into Japanese), and represents some abstract meanings. Many Japanese words are written in only kanji characters.[1] For example, Japanese words 選択, 選別, 選考, and 選手 share the kanji character 選, which represents the meaning of “to select,” and as a result they are somewhat semantically related. People familiar with Japanese easily recognize that these words are semantically related, even though they do not know the exact meanings of these words. This feature of kanji characters indicates that, when word-based DSMs are applied to Japanese, the “words as semantic units” assumption fails to exploit the useful semantic information of kanji characters.

This merit of kanji characters is particularly beneficial for less frequent or rarely used words. Given that data sparseness is one of the most serious problems with DSMs, the semantic information involved in kanji characters can contribute to the improvement of quality of infrequent word vectors. Furthermore, the high lexical productivity of Japanese language strengthens the need to incorporate the semantic information of kanji characters into DSMs. Japanese words are easy to create by combining kanji characters. For example, by combining the kanji 選 (*to select/choose*) and 語 (*word*), we can easily create a novel word 選語 (*to select/choose words*). People familiar with Japanese easily interpret the meaning of this pseudo-word as “to select/choose words.” However, word-based DSMs cannot deal with this pseudo-word because it is not a word and thus does not appear in any corpus.

In this paper, therefore, we propose a method for exploiting the semantic information of kanji characters for constructing Japanese word vectors in DSMs as the first step toward a full-fledged kanji-based DSM (henceforth, kanji-DSM). In the proposed method, the semantic representations of kanji characters (i.e, kanji vectors) are constructed first using the techniques of traditional word-based DSMs (henceforth, word-DSM), and then word vectors are computed by combining the vectors of constituent kanji

[1] The Japanese language also has phonograms *hiragana* and *katakana*, collectively known as *kana*, and all Japanese words are written in kanji and kana characters.

characters. We adapt a number of composition methods for computing word vectors from kanji vectors, and we show that the proposed kanji-DSM is effective in constructing Japanese word vectors, especially word vectors for less frequent words, without word distributional statistics.

2. Related Work

Some previous NLP studies exploit the meanings of kanji characters for semantic tasks such as document indexing (Ogawa and Iwasaki, 1995). However, at least to our knowledge, no studies on DSMs have addressed the semantic information of kanji characters. Very recently, Inohara and Ueno (2014) applied latent semantic analysis (henceforth, LSA) to Japanese kanji characters in order to examine whether kanji characters with the same radical (i.e., the left component of kanji characters) are more semantically similar to each other than those with different radicals. However, they did not attempt to construct word vectors from kanji vectors.

As we mentioned in the introduction, English and other phonogram-based languages also have sublexical semantic components, namely derivational morphemes. Derivational morphemes represent a semantic function over a word, by which new words are constructed from existing stem words. For example, the addition of the morpheme *-less* to a noun creates a new adjective with the sense of lacking or being without the thing or quality expressed by that noun. Recently, Lazaridou et al. (2013) and Padó et al. (2013) applied vector composition methods in DSMs to the process of morphological derivation. The basic idea underlying their studies is similar to our study; the vector of the derived form (e.g., *shameless*) is computed by combining the vector of a stem word *shame* with the vector of a derivational morpheme *-less*. However, our study differs from their studies; a kanji-DSM can construct word vectors without any word vectors, but their morphology-based DSM constructs new word vectors using existing word vectors within the same semantic space.

3. Kanji-based DSM

The kanji-based DSM we propose in this paper comprises the following two steps.

1. Kanji vectors are constructed using the techniques of traditional DSMs.

2. Word vectors are constructed from the vectors of their constituent kanji characters.

In this section, we explain these two steps in detail.

3.1. Constructing kanji vectors

The method for constructing kanji vectors comprises the following three steps.

1. Initial matrix construction: n kanji characters in a given corpus are represented as k-dimensional initial vectors whose elements are frequencies in a linguistic context. As a result, an n by k matrix $\boldsymbol{A}$ is constructed using n kanji vectors as rows.

2. Weighting: The elements of the matrix $\boldsymbol{A}$ are weighted.

3. Smoothing: The dimension of $\boldsymbol{A}$'s row vectors is reduced from k to m.

As a result, an m-dimensional semantic space including n kanji characters is generated. The last step of smoothing is optional; in this paper, we do not smooth the matrix because the dimension of kanji vectors is not high enough for smoothing to be effective, and thus $m = k$.

For initial matrix construction in Step 1, we use kanji characters as contexts. In other words, we count the number of times a target kanji character occurs with other kanji characters within a certain context. In this paper, two types of contexts are considered: a *within-word* (or intra-word) context and a *between-word* (or inter-word) context. In the case of within-word contexts, an element a_{ij} of an initial matrix $\boldsymbol{A_{within}}$ is the number of times a kanji c_i cooccurs with a kanji c_j within the same words. On the other hand, in the case of between-word contexts, an element a_{ij} of $\boldsymbol{A_{between}}$ is the number of times a kanji c_i in a word cooccurs with a kanji c_j in other words within a "window" spanning a certain number of words. For example, given the sentence "民主党 / が / 選挙 / で / 敗北 / する (*The Democratic Party loses in the election*)," 民 – 主, 民 – 党, and 選 – 挙 are counted as one cooccurrence, but 民 – 選, 民 – 挙, and 民 – 敗 are not in the within-word counting scheme. In the between-word counting scheme, the reverse is true; 民 – 選, 民 – 挙, and 民 – 敗 are counted as one cooccurrence, but 民 – 主, 民 – 党, and 選 – 挙 are not.

Using these two counting methods, we consider the following four initial matrices:

$$\text{within-word:} \quad \boldsymbol{A} = \boldsymbol{A_{within}} \qquad (1)$$

$$\text{between-word:} \quad \boldsymbol{A} = \boldsymbol{A_{between}} \qquad (2)$$

$$\text{Addition:} \quad \boldsymbol{A} = \boldsymbol{A_{within}} + \boldsymbol{A_{between}} \qquad (3)$$

$$\text{Concatenation:} \quad \boldsymbol{A} = (\boldsymbol{A_{within}} | \boldsymbol{A_{between}}) \qquad (4)$$

In Step 2, two popular weighting schemes are used: tf-idf weighting (Landauer and Dumais, 1997) and PPMI weighting (Bullinaria and Levy, 2007; Recchia and Jones, 2009). LSA often uses the tf-idf weighting scheme, and in this paper we use the product of the logarithm of kanji frequency and the entropy (Quesada, 2007). The PPMI weighting scheme is widely used and generally achieves good performance. It is based on the pointwise mutual information (PMI) and replaces negative PMI values with zero.

3.2. Constructing word vectors

The vector for a word can be computed by applying vector composition methods to its constituent kanji vectors. For example, the vector for the word 選択 is computed by combining the two kanji vectors of 選 and 択.

Recently, much progress has been made in composition methods for DSMs (Baroni et al., in press; Clark, 2012). Among a variety of composition methods, a class of methods based on simple component-wise operations (Mitchell and Lapata, 2010) is the most popular. In this paper, we use the following four composition methods of this class.

When a word vector $\boldsymbol{w}$ is constructed from N kanji vectors $\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N$, each element w_i of the word vector $\boldsymbol{w}$ is computed by these composition methods as follows:

- Centroid (Cent): $w_i = \frac{\sum_{j=1}^{N} p_{ji}}{N}$
- Multiplication (Mul): $w_i = \prod_{j=1}^{N} p_{ji}$
- Averaged multiplication (AveMul): $w_i = \sqrt[(N)]{\prod_{j=1}^{N} p_{ji}}$
- Min (Min): $w_i = \min_{j} p_{ji}$

The centroid method is widely used for vector composition in DSMs. The multiplication method uses component-wise multiplication for vector composition. This method has been demonstrated to achieve better performance than the additive class of composition methods such as centroid (Mitchell and Lapata, 2010; Utsumi, 2014). The averaged multiplication method can be seen as a multiplicative version of the centroid method; each component of a vector is calculated as a geometric mean, rather than a simple multiplication. The min method is originally devised in this paper to offer only a function of downplaying irrelevant dimensions, although the multiplicative model also functions to highlight dimensions relevant to both vectors.

We also use two composition methods for combining two input vectors, where the composed vector $\boldsymbol{w}$ is a weighted sum or product of two vectors $\boldsymbol{u}$ and $\boldsymbol{v}$.

- Weighted addition (WAdd): $\boldsymbol{w} = \alpha \boldsymbol{u} + \beta \boldsymbol{v}$
- Weighted multiplication (WMul): $\boldsymbol{w} = \boldsymbol{u}^{\alpha} \odot \boldsymbol{v}^{\beta}$

Note that the symbol $\odot$ expresses that two vectors are multiplied component-wise, and $\boldsymbol{u}^{\alpha}$ denotes that each element of the vector $\boldsymbol{u}$ is raised to the α-th power. When a word comprises more than two kanji characters, all kanji vectors are combined in order from the first two kanji characters.

More complex methods — e.g., the full additive model (Zanzotto et al., 2010), linear transformations (Baroni et al., in press) and deep neural networks (Socher et al., 2012) — that require training data (i.e., in the case of kanji-DSMs, word vectors in word-DSMs constructed from the corpus) are not used in this paper. One serious problem in applying these methods to kanji-DSMs is that training word vectors and its constituent kanji vectors lie in different vector spaces. Of course, these methods can map onto different spaces from those of input vectors, but it does not imply that these mappings are semantically appropriate, given that dimensions (or features) of different spaces may have different semantics. One possible approach to handling this problem would be that the same linguistic context (e.g., words) is employed to construct the initial matrix for both kanji and word vectors. The analysis by this approach or the exploration of other approaches is left for further research.

4. Evaluation

4.1. Method

To evaluate the performance of the kanji-DSM proposed in this paper, we conduct an experiment using the task of synonym identification. Synonym identification is the task in which for a given target word w^T, its synonymous words are predicted. The computer's prediction using DSMs is determined by computing the cosine similarity between the target word and all other words in a given word list, and retrieving top r words similar to the target word as a candidate of synonyms.

The score of this task is computed as the minimum number r_i of retrieved words that include a synonymous word w_i^S for a target word r_i^T given as the test data. In other words, it corresponds to the rank of a synonymous word w_i^S when all words are sorted in descending order of cosine similarity to the target word w_i^T. The total performance for the test data (i.e., all target-synonym pairs) is measured by the median $\tilde{r}$ of the task score r_i (i.e., *the median rank*) over all target-synonym pairs.

4.2. Test data

As test data, 200 pairs of Japanese synonymous words are selected from the Japanese thesaurus "Nihongo Dai-Thesaurus" so that all words comprise two kanji characters, e.g., a pair of 威嚇 and 脅迫 (both words mean "to threaten"). For each pair of words, two test pairs of a target word and its synonymous word are generated (e.g., a pair of a target word 威嚇 and its synonym 脅迫, and a pair of a target word 脅迫 and its synonym 威嚇). As a result, 400 target-synonym pairs are used in the evaluation.

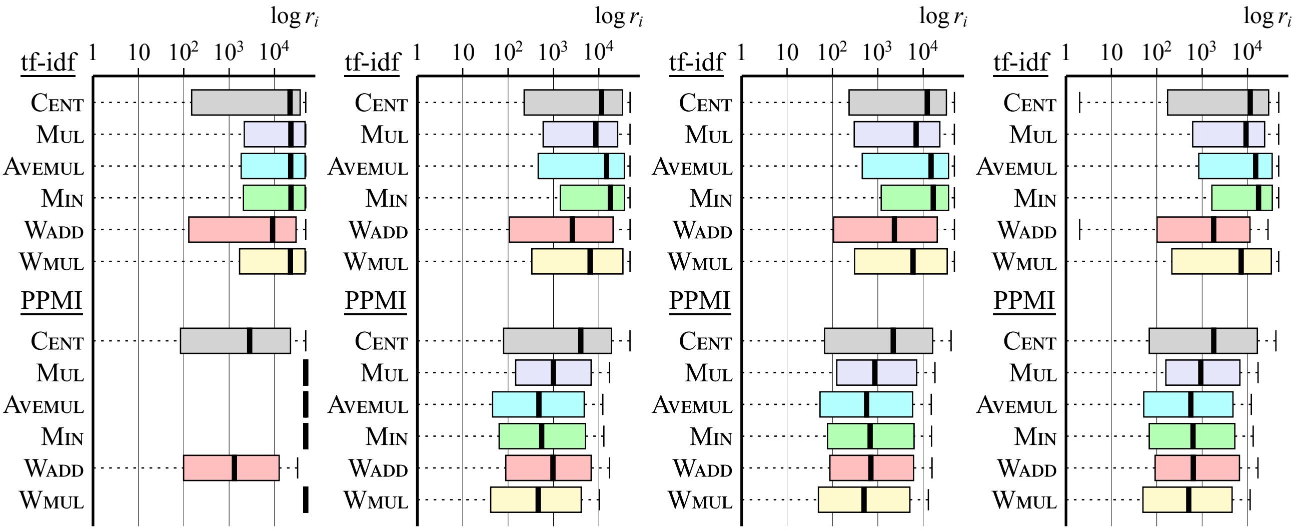
4.3. Corpus and DSMs

The corpus used for constructing kanji-DSMs and word-DSMs is six years' worth of Japanese Mainichi newspaper articles published in 2006 through 2011. The corpus contains 990,381 documents and 41,202,145 words. The vocabulary consists of 75,043 different words and 3,569 different kanji characters.

Using this corpus, we construct eight kanji-DSMs (i.e., 4 initial matrices × 2 weighting methods), and apply six composition methods to each of the kanji-DSMs. Hence, 48 (=8×6) different results are obtained in the evaluation. A context window of size 1 (i.e., one word on either side of the target word) is used in the between-word counting method for initial matrix construction. The justification of using this window size will be discussed later in this section.

To evaluate the relative performance of the kanji-DSMs, we also construct four word-DSMs using two initial matrices and two weighting schemes. An initial matrix is a word-document matrix whose elements are frequencies of a word in a document, or a word-word matrix whose elements are frequencies of word cooccurrences within the same context window as the kanji-DSMs. The same weighting schemes (i.e., tf-idf and PPMI) are used in word-DSMs. All matrices are smoothed by singular value decomposition. Note that LSA corresponds to the word-DSM with a word-document matrix and a tf-idf weighting.

To compute the test score r_i of a given target-synonym pair, we must compute the cosine similarity between the



(a) Within-word matrix (b) Between-word matrix (c) Addition matrix (d) Concatenation matrix

Figure 2: Boxplots of the rank of cosine similarity for synonymous word pairs. The results of all composition methods for all kanji-DSMs are shown.

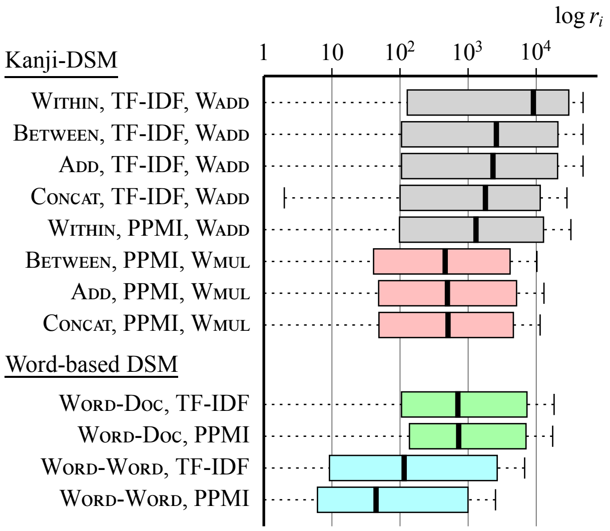


Figure 1: Boxplots of the rank of cosine similarity for synonymous word pairs computed in kanji-DSMs and word-DSMs. For kanji-DSMs, only the best result among the six composition methods is presented.

target word and all other words. As mentioned above, the corpus contains 75,043 different words, but some of them include kana (i.e., phonetic-based) characters. To make a fair comparison of the performance between kanji-DSMs and word-DSMs, we exclude these words from the similarity computation. As a result, 48,798 words that include only kanji characters are used for computation. [2]

To estimate the parameters of Wadd and Wmul composition methods for kanji-DSMs and the dimensionality of word-DSMs, we use a leave-one-out cross-validation procedure. The parameters α and β for Wadd and Wmul composition methods are determined over real numbers ranging between 0.1 and 1.0 in steps of 0.1. The number of dimensions for word-DSMs is determined over integers ranging from 100 to 1,500 in steps of 100.

4.4. Result

Overall performance of kanji-DSM Figure 1 shows the results of synonym identification for kanji-DSMs and word-DSMs. The three labels for kanji-DSMs (e.g.,Within, TF-IDF, Wadd) respectively denote an initial matrix, a weighting scheme, and a vector composition method. Similarly, the two labels for word-DSMs (e.g.,Word-Doc, TF-IDF) denote an initial matrix and a weighting scheme. In Figure 1, only the best result among the six composition methods is shown for each kanji-DSM. The results of all composition methods for kanji-DSMs are presented in Figure 2. Note that the line inside these boxplots denotes the median rank, and the left and right side of the box respectively denote the first quartile rank (i.e., the median of the lower half of the data) and third quartile rank (i.e., the median of the upper half of the data).

The most important result shown in Figure 1 is that some kanji-DSMs (indicated by red boxplots) perform better than the word-DSMs with a word-document matrix including LSA (indicated by green boxplots), although outperformed by the word-DSMs with a word-word matrix (indicated by blue boxplots). This result indicates that the kanji-DSM proposed in this paper has an ability to represent the meanings of words, and exploiting the semantic contents of kanji characters is beneficial in constructing word vectors for the

[2] We also conducted an evaluation experiment by adopting a different approach in handling kana characters, in which we assigned to each kana character a random vector generated from a uniform distribution between 0 and 1 so that it has the same degree of sparseness as kanji vectors. The obtained result did not greatly differ from the results reported in this paper, but we decided not to use this approach in this paper because it is difficult to guarantee that random vectors are truly the same degree of sparseness as kanji vectors.

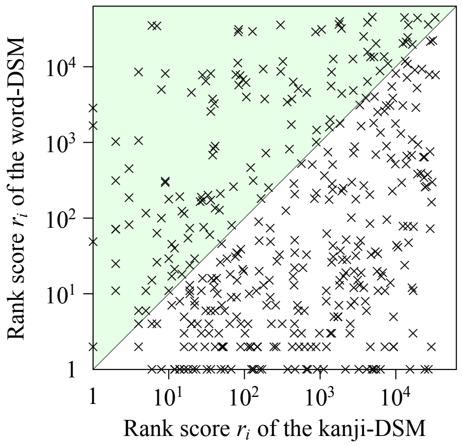


Figure 3: Scatterplot of the rank score r_i between the kanji-DSM (Between, PPMI and Wmul) and the word-DSM (Word-Word and PPMI) that achieve the best performance for all 400 target-synonym pairs. The points that lie above the diagonal line (i.e., in the region colored by green) indicate the target-synonym pairs for which the kanji-DSM outperforms the word-DSM.

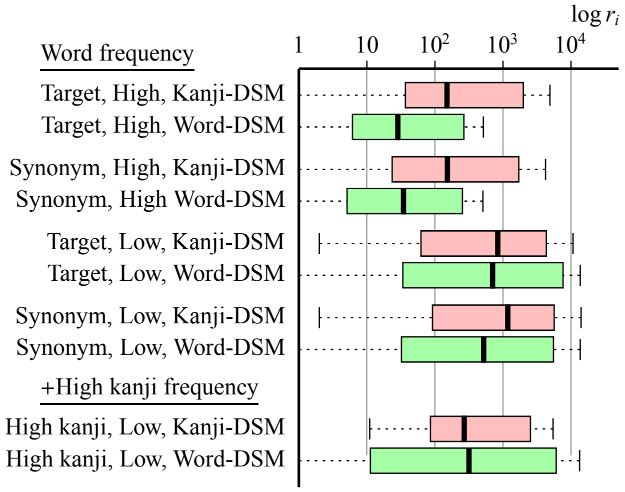


Figure 4: Word frequency effects on the performance of the best kanji-DSM and word-DSM.

Japanese language.

Comparison among different kanji-DSMs shows that the PPMI weighting is better than the tf-idf weighting. This finding is not surprising; such the superiority is also demonstrated in the case of the original word-DSMs (Mitchell and Lapata, 2010; Utsumi, 2014). In addition, the multiplicative class of composition algorithms is appropriate for PPMI-based kanji-DSMs, although the additive class is appropriate for tf-idf-based kanji-DSMs. Concerning a method for initial matrix construction, no clear differences are observed, except that the within-word matrix performs worse than the other matrices. This difference is also reasonable because the kanji-kanji matrices generated by within-word counting are more sparse, and thus may involve less semantic information.

For what case does kanji-DSM outperform word-DSM? The evaluation experiment demonstrates that the kanji-DSM does not outperform the word-word-matrix-based word-DSM. However, it does not necessarily indicate that in *all* cases the kanji-DSM is worse than the word-DSM. Indeed, as shown in Figure 3, a considerable number of synonym pairs are better predicted by the kanji-DSM than by the word-DSM, thus suggesting that the kanji-DSM outperforms the word-DSM in a not negligible number of cases.

One question that arises here is for what case the kanji-DSM outperforms the word-DSM. To answer this question, we focus on the word frequency of target and synonym words. It is motivated by our intuitive assumption that the kanji-DSM can resolve the data sparseness problem of word-DSMs and improve the quality of word vectors for infrequent or novel words. If our assumption is right, the kanji-DSM would surpass the word-DSM in the case of target-synonym pairs including less frequent words.

For this purpose, we choose the 100 most and least frequent words from among all words included in target-synonym pairs, and generate four groups of target-synonym pairs according to whether the target or synonym of a pair is frequent or infrequent. Figure 4 shows the median rank of the best kanji-DSM and word-DSM for the four groups of pairs. Overall, as predicted, both kanji-DSM and word-DSM perform better for target-synonym pairs with highly frequent words (indicated by the first four boxplots) than for those with less frequent words (indicated by the next four boxplots). When target-synonym pairs contain frequent words, the word-DSM (denoted by green boxplots) is superior to the kanji-DSM (denoted by red boxplots). Furthermore, when target-synonym pairs contain less frequent words, the kanji-DSM does still not surpass the word-DSM, although the difference of median rank is smaller than in the case of frequent pairs.

This result appears not to be consistent with our assumption that kanji-DSMs particularly contribute to the improvement of vector representation of less frequent words. However, there is a possibility that less frequent words also comprise infrequent kanji characters. In this case, kanji-DSMs also suffer from the same data sparseness problem as word-DSMs. For example, the word 廉価 (*cheap price*) included in our dataset is infrequent (e.g., the frequency of 廉価 is below one hundredth of the frequency of its synonymous word 安価 in our corpus), and the kanji character 廉 is also infrequent because it is rarely used in other words. Rather, kanji-DSMs are beneficial to novel words created by combining frequent kanji characters, which is true in almost all the cases of word formation. Therefore, we choose infrequent words comprising only frequent kanji characters (i.e., the minimum frequency of constituent kanji characters is high) from among the 100 infrequent words, and examine the performance for those infrequent words. The result is shown by the last two boxplots of Figure 4. As predicted, the kanji-DSM (whose median rank is 269.5) outperforms the word-DSM (whose median rank is 316.5), thus suggesting that kanji-DSMs have merit in improving the quality of less frequent word vectors.

Size matters In the evaluation experiment, the context window of size 1 is used for constructing both kanji-DSMs and word-DSMs. This is motivated by the empirical finding that small context windows are more appropriate for

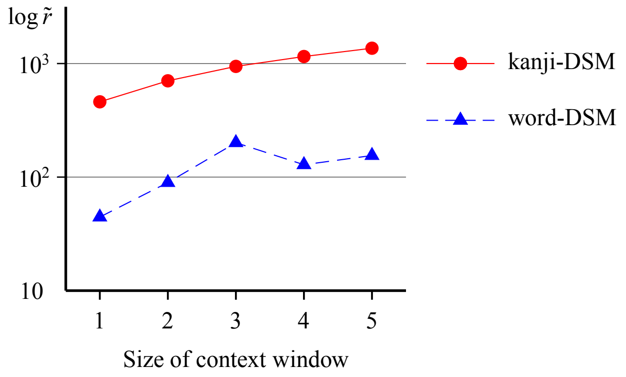


Figure 5: Median rank of the best kanji-DSM and word-DSM as a function of context window size.

representing the synonym relation, or paradigmatic relation in general (Peirsman et al., 2008; Utsumi, 2010). To confirm that it is also the case for kanji-DSMs, we examine the performance of the synonym identification task using kanji-DSMs with larger windows. Figure 5 shows the median ranks of kanji-DSMs with larger windows, together with the result of word-DSMs. The median ranks depicted in Figure 5 are obtained using the kanji-DSMs (Between, PPMI and Wmul) and word-DSMs (Word-Word and PPMI) that yield the best performance in Figure 1. The performance of the kanji-DSM declines as the window size becomes larger, just as the performance of the word-DSM does; both DSMs achieve the best result when the window size is 1. (Note that the graph of median rank rises as the window size increases, because a higher median rank implies worse performance.)

However, the superiority of smaller window size does not necessarily hold for other semantic relations than synonymy. For example, a syntagmatic relation (i.e., a relation between two words that cooccur more often than would be expected by chance) is better represented by DSMs with larger context windows (Peirsman et al., 2008; Utsumi, 2010). In this paper, we use only the synonym identification task to evaluate kanji-DSMs, but as future work we need to examine the ability of kanji-DSMs to compute the semantic relatedness in general, using other tasks such as association prediction and semantic categorization.

5. Concluding Remarks

In this paper, we have proposed one implementation of kanji-DSMs that exploits semantic information involved in kanji characters for constructing Japanese word vectors. We have also demonstrated that the proposed kanji-DSM outperforms LSA in the task of synonym identification, and also surpasses the best score word-DSM in vectorizing infrequent words comprising only frequent kanji characters.

Although generally not outperforming the best score word-DSM, the kanji-DSM has a number of advantages over word-DSMs. First, as mentioned in the introduction, the kanji-DSM can deal with novel or infrequent words in the same way as other words, although word-DSMs cannot represent the meaning of words that do not appear in a corpus (nor generate an appropriate representation for unfamiliar or infrequent words). Second, the kanji-DSM does not suffer from errors arising from a morphological analyzer. In word-DSMs, a morphological analyzer must be used to segment sentences into words to construct an initial matrix because Japanese sentences are written without spaces between words. On the other hand, the kanji-DSM does not require such the segmentation. Finally, the kanji-DSM can offer a computational framework of cognitive modeling for Japanese lexical processing in that a number of psycholinguistic studies (Miwa et al., 2014) have shown that kanji characters have an inevitable effect on lexical processing of Japanese words.

At the same time, the kanji-DSM has some limitations. One serious limitation is that it cannot deal with words involving no kanji characters. For example, loan words (i.e., words borrowed from foreign languages) are generally written in katakana (except the words of Chinese origin), e.g., コンピュータ (*computer*), カメラ (*camera*). Another limitation is that many kanji characters are semantically highly ambiguous because many different words share the same kanji character, and thus it may be difficult to separate a proper meaning by simple vector composition.

Taken together, it is concluded that a kanji-DSM and a word-DSM are complementary, not competitive, models. To make use of the advantages and overcome the limitations, kanji-DSMs should be integrated with word-DSMs, which is an important issue for future research. Furthermore, to improve the quality of kanji-DSMs, it would also be vital for future research to take account of the semantic relation holding between constituent kanji characters of a word in vector composition. For example, 選択 (*to select*) is constructed by coordinating two semantically similar characters 選 (*select*) and 択 (*choose from options*), while 欠点 (*weak point, shortcoming*) is formed by modifying or constraining the meaning of 点 (*point*) by 欠 (*weak*). Such the differences should be reflected in the choice of composition methods or other processing factors in kanji-DSMs.

6. Acknowledgments

This study was supported in part by Grants-in-Aid for Scientific Research B (No.23300098) from the Japan Society for the Promotion of Science. I would like to thank three anonymous reviewers for their insightful comments and suggestions.

7. References

Marco Baroni, Raffaella Bernardi, and Roberto Zamparelli. in press. Frege in space: A program for compositional distributional semantics. *Linguistic Issues in Language Technologies*.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods*, 39(3):510–526.

Stephen Clark. 2012. Vector space models of lexical meaning. In Shalom Lappin and Chris Fox, editors, *Handbook of Contemporary Semantics, 2nd Edition*. Wiley-Blackwell. in press.

Keisuke Inohara and Taiji Ueno. 2014. Contribution of sublexical information to word meaning: An objective approach from the latent semantic analysis and from the corpus analysis on predicates. Unpublished manuscript.

Michael N. Jones, Walter Kintsch, and Douglas J.K. Mewhort. 2006. High-dimensional semantic space accounts of priming. *Journal of Memory and Language*, 55:534–552.

Thomas K. Landauer and Susan T. Dumais. 1997. A solution to Plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review*, 104:211–240.

Thomas K. Landauer, Danielle S. McNamara, Simon Dennis, and Walter Kintsch. 2007. *Handbook of Latent Semantic Analysis*. Lawrence Erlbaum Associates, Mahwah, NJ.

Angeliki Lazaridou, Marco Marelli, Roberto Zamparelli, and Marco Baroni. 2013. Compositional-ly derived representations of morphologically complex words in distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*, pages 1517–1526.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive Science*, 34:1388–1429.

Koji Miwa, Gary Libben, Ton Dijkstra, and Harald Baayen. 2014. The time-course of lexical activation in Japanese morphographic word recognition: Evidence for a character-driven processing model. *The Quarterly Journal of Experimental Psychology*, 67(1):79–113.

Yasushi Ogawa and Masajirou Iwasaki. 1995. A new character-based indexing method using frequency data for Japanese documents. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, pages 121–129.

Sebastian Padó, Jan Šnajder, and Britta Zeller. 2013. Derivational smoothing for syntactic distributional semantics. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL2013)*, pages 731–735.

Yves Peirsman, Kris Heylen, and Dirk Geeraerts. 2008. Size matters: Tight and loose context definitions in English word space models. In *Proceedings of the ESSLLI Workshop on Distributional Lexical Semantics*, pages 34–41.

Jose Quesada. 2007. Creating your own LSA spaces. In T.K. Landauer, D.S. McNamara, S. Dennis, and W. Kintsch, editors, *Handbook of Latent Semantic Analysis*, pages 71–85. Lawrence Erlbaum Associates, Mahwah, NJ.

Gabriel Recchia and Michael N. Jones. 2009. More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. *Behavior Research Methods*, 41:647–656.

Richard Socher, Brody Huval, Christopher D. Manning, and Andrew T. Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211.

Peter D. Turney and Patrick Pantel. 2010. From frequency to meaning: Vector space models of semantics. *Journal of Artificial Intelligence Research*, 37:141–188.

Akira Utsumi. 2010. Exploring the relationship between semantic spaces and semantic relations. In *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC2010)*, pages 257–262.

Akira Utsumi. 2011. Computational exploration of metaphor comprehension processes using a semantic space model. *Cognitive Science*, 35(2):251–296.

Akira Utsumi. 2014. A semantic space approach to the computational semantics of noun compounds. *Natural Language Engineering*, 20(2):185–234.

Dominic Widdows. 2004. *Geometry and Meaning*. CSLI Publications.

Fabio M. Zanzotto, Ioannis Korkontzelos, Francesca Fallucchi, and Suresh Manandhar. 2010. Estimating linear models for compositional distributional semantics. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-2010)*, pages 1263–1271.