Substring Frequency Features for Segmentation of Japanese Katakana Words with Unlabeled Corpora

Yoshinari Fujinuma*
Computer Science
University of Colorado
Boulder, CO
Yoshinari.Fujinuma@colorado.edu

Alvin C. Grissom II
Mathematics and Computer Science
Ursinus College
Collegeville, PA
agrissom@ursinus.edu

Abstract

Word segmentation is crucial in natural language processing tasks for unsegmented languages. In Japanese, many out-of-vocabulary words appear in the phonetic syllabary katakana, making segmentation more difficult due to the lack of clues found in mixed script settings. In this paper, we propose a straightforward approach based on a variant of tf-idf and apply it to the problem of word segmentation in Japanese. Even though our method uses only an unlabeled corpus, experimental results show that it achieves performance comparable to existing methods that use manually labeled corpora. Furthermore, it improves performance of simple word segmentation models trained on a manually labeled corpus.

1 Introduction

In languages without explicit segmentation, word segmentation is a crucial step of natural language processing tasks. In Japanese, this problem is less severe than in Chinese because of the existence of three different scripts: *hiragana*, *katakana*, and *kanji*, which are Chinese characters.[1] However, katakana words are known to degrade word segmentation performance because of out-of-vocabulary (OOV) words which do not appear manually segmented corpora (Nakazawa et al., 2005; Kaji and Kitsuregawa, 2011). Creation of new words is common in Japanese; around 20% of the katakana words in newspaper articles are OOV words (Breen, 2009). For example, some katakana compound loanwords are not transliterated but rather "Japanized" (e.g., ガソリンスタンド *gasorinsutando* "gasoline stand", meaning "gas station" in English) or abbreviated (e.g., スマートフォンケース *sumātofonkēsu* ("smartphone case"), which is abbreviated as スマホケース *sumahokēsu*). Abbreviations may also undergo phonetic and corresponding orthographic changes, as in the case of スマートフォン *sumātofon* ("smartphone"), which, in the abbreviated term, shortens the long vowel *ā* to *a*, and replaces フォ *fo* with ホ *ho*. This change is then propagated to compound words, such as スマホケース *sumahokēsu* ("smartphone case"). Word segmentation of compound words is important for improving results in downstream tasks, such as information retrieval (Braschler and Ripplinger, 2004; Alfonseca et al., 2008), machine translation (Koehn and Knight, 2003), and information extraction from microblogs (Bansal et al., 2015).

Hagiwara and Sekine (2013) incorporated an English corpus by projecting Japanese transliterations to words from an English corpus; however, loanwords that are not transliterated (such as *sumaho* for "smartphone") cannot be segmented by the use of an English corpus alone. We investigate a more efficient use of a Japanese corpus by incorporating a variant of the well-known tf-idf weighting scheme (Salton and Buckley, 1988), which we refer to as *term frequency-inverse substring frequency* or **tf-issf**. The core idea of our approach[2] is to assign scores based on the likelihood that a given katakana substring is a word token, using only statistics from an unlabeled corpus.

[*] Part of the work was done while the first author was at Amazon Japan.

[1] Hiragana and katakana are the two distinct character sets representing the same Japanese sounds. These two character sets are used for different purposes, with katakana typically used for transliterations of loanwords. Kanji are typically used for nouns.

[2] Our code is available at https://www.github.com/akkikiki/katakana_segmentation.

Proceedings of the The 8th International Joint Conference on Natural Language Processing, pages 74–79,
Taipei, Taiwan, November 27 – December 1, 2017 ©2017 AFNLP

Our contributions are as follows:

1. We show that a word segmentation model using tf-issf alone outperforms a previously proposed frequency-based method and that it produces comparable results to various Japanese tokenizers.
2. We show that tf-issf improves the F1-score of word segmentation models trained on manually labeled data by more than 5%.

2 Proposed Method

In this section, we describe the katakana word segmentation problem and our approach to it.

2.1 Term Frequency-Inverse Substring Frequency

Let S be a sequence of katakana characters, Y be the set of all possible segmentations, $y \in Y$ be a possible segmentation, and y_i be a substring of y. Then, for instance, $y = y_1 y_2 ... y_n$ is a possible word segmentation of S with n segments.

We now propose a method to segment katakana OOV words. Our approach, *term frequency-inverse substring frequency* (**tf-issf**), is a variant of the tf-idf weighting method, which computes a score for each candidate segmentation. We calculate the score of a katakana string y_i with

$$\text{tf-issf}(y_i) = \frac{tf(y_i)}{sf(y_i)}, \qquad (1)$$

where $tf(y_i)$ is the number of occurrences of y_i as a katakana term in a corpus and $sf(y_i)$ is the number of unique katakana terms that have y_i as a substring. We regard consecutive katakana characters as a single katakana term when computing tf-issf.

We compute the product of tf-issf scores over a string to score the segmentation

$$Score(y) = \prod_{y_i \in y} \text{tf-issf}(y_i), \qquad (2)$$

and choose the optimal segmentation y^* with

$$y^* = \arg\max_{y \in Y} Score(y|S). \qquad (3)$$

Intuitively, if a string appears frequently as a word substring, we treat it as a meaningless sequence.[3] While substrings of consecutive katakana can, in principle, be a meaningful character n-gram, this rarely occurs, and tf-issf successfully penalizes the score of such sequences of characters.

ID	Notation	Feature
1	y_i	unigram
2	y_{i-1}, y_i	bigram
3	length(y_i)	num. of characters in y_i

Table 1: Features used for the structured perceptron.

Figure 1 shows an example of a word lattice for the loan compound word "smartphone case" with the desired segmentation path in bold. When building a lattice, we only create a node for a substring that appears as a term in the unlabeled corpus and does not start with a small katakana letter[4] or a prolonged sound mark "ー", as such characters are rarely the first character in a word. Including terms or consecutive katakana characters from an unlabeled corpus reduces the number of OOV words.

2.2 Structured Perceptron with tf-issf

To incorporate manually labeled data and to compare with other supervised Japanese tokenizers, we use the structured perceptron (Collins, 2002). This model represents the score as

$$Score(y) = \mathbf{w} \cdot \phi(y), \qquad (4)$$

where $\phi(y)$ is a feature function and $\mathbf{w}$ is a weight vector. Features used in the structured perceptron are shown in Table 1. We use the surface-level features used by Kaji and Kitsuregawa (2011) and decode with the Viterbi algorithm. We incorporate tf-issf into the structured perceptron as the initial feature weight for unigrams instead of initializing the weight vector to 0.[5] Specifically, we use $\log(\text{tf-issf}(y_i) + 1)$ for the initial weights to avoid overemphasizing the tf-issf value (Kaji and Kitsuregawa, 2011). In this way, we can directly adjust tf-issf values using a manually labeled corpus. Unlike the approach of Xiao et al. (2002), which uses tf-idf to resolve segmentation ambiguities in Chinese, we regard each katakana term as one document to compute its inverse document (substring) frequency.

[3] A typical example is a single character substring. However, it is possible for single-character substrings could be word tokens.

[4] Such letters are ァ *a*, ィ *i*, ゥ *u*, ェ *e*, ォ *o*, ヵ *ka*, ヶ *ke*, ッ *tsu*, ャ *ya*, ュ *yu*, ョ *yo*, and ヮ *wa*.

[5] We also attempt to incorporate tf-issf as an individual feature, but this does not improve the segmentation results.

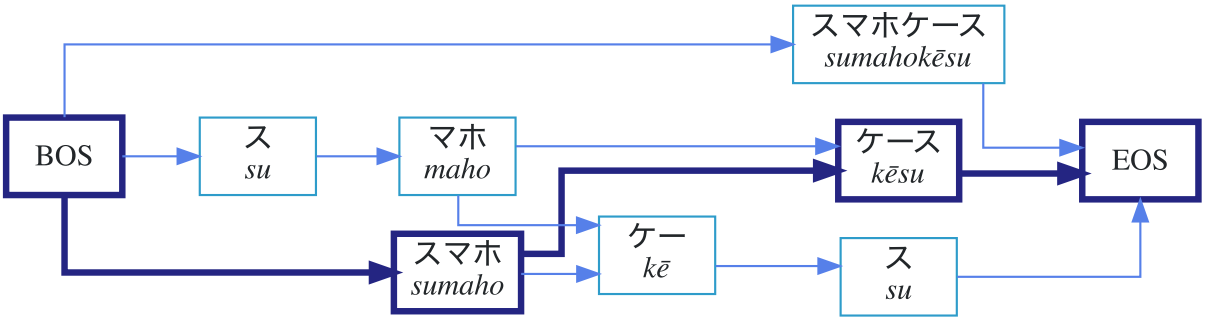

Figure 1: An example lattice for a katakana word segmentation. We use the Viterbi algorithm to find the optimal segmentation from the beginning of the string (BOS) to the end (EOS), shown by the bold edges and nodes. Only those katakana substrings which exist in the training corpus as words are considered. This example produces the correct segmentation, スマホ / ケース *sumaho / kēsu* ("smartphone case").

3 Experiments

We now describe our experiments. We run our proposed method under two different settings: 1) using only an unlabeled corpus (UNLABEL), and 2) using both an unlabeled corpus and a labeled corpus (BOTH). For the first experiment, we establish a baseline result using an approach proposed by Nakazawa et al. (2005) and compare this with using tf-issf alone. We conduct an experiment in the second setting to compare with other supervised approaches, including Japanese tokenizers.

3.1 Dataset

We compute the tf-issf value for each katakana substring using all of 2015 Japanese Wikipedia as an unlabeled training corpus. This consists of $1,937,006$ unique katakana terms.

Following Hagiwara and Sekine (2013), we test on both the general domain and on a domain with more OOV words. We use the Balanced Corpus of Contemporary Written Japanese (BCCWJ) (Maekawa et al., 2014) core data with $40,827$ katakana entries as the general domain test data. We use 3-fold cross-validation to train a structured perceptron classifier. To test on a more challenging domain with more OOV words (Saito et al., 2014) and even fewer space boundaries (Bansal et al., 2015), we also ask an annotator to label Twitter hashtags that *only* use katakana. We gather $273,711$ tweets with at least one hashtag from September 25, 2015 to October 28, 2015 using the Twitter Streaming API.[6] This provides a total of $4,863$ unique katakana hashtags, of which $1,251$ are observed in BCCWJ core data. We filter out duplicate hashtags because the Twitter Streaming API collects a set of sample tweets that are biased compared with the overall tweet stream (Morstatter et al., 2013). We follow the BCCWJ annotation guidelines (Ogura et al., 2011) to conduct the annotation.[7]

3.2 Baselines

We follow previous work and use a frequency-based method as the baseline (Nakazawa et al., 2005; Kaji and Kitsuregawa, 2011):

$$y^* = \arg\max_{y \in Y} \frac{\left(\prod_{i=1}^{n} tf(y_i)\right)^{\frac{1}{n}}}{\frac{C}{N^l} + \alpha} \tag{5}$$

where l is the average length of all segmented substrings. Following Nakazawa et al. (2005) and Kaji and Kitsuregawa (2011), we set the hyperparameters to $C = 2500$, $N = 4$, and $\alpha = 0.7$. In addition, we filter out segmentations that have a segment starting with a small katakana letter or a prolonged sound mark "ー". The key difference between the baseline and tf-issf is that the length of a segmented substring is considered in the baseline method. An advantage of tf-issf over the baseline is that hyperparameters are not required.

Unsupervised segmentation (Goldwater et al., 2006; Mochihashi et al., 2009) can also be applied to katakana word segmentation; however, doing so this on a large corpus is still challenging (Chen, 2013). Our work focuses on fast and scalable frequency-based methods.

We compare the performance of the word segmentation model trained with the structured per-

[6] https://dev.twitter.com/streaming/overview

[7] In addition, we stipulate that we always split transliterated compound words according to their spaces when they are back-transliterated to their original language.

		BCCWJ				Twitter Hashtags			
	Method	**WER**	**P**	**R**	**F1**	**WER**	**P**	**R**	**F1**
Frequency	Baseline	.174	.865	.890	.878	.576	.578	.716	.640
(UNLABEL)	tf-issf	**.119**	**.913**	**.907**	**.910**	**.312**	**.758**	**.784**	**.771**
Structured	unigram	.023	.979	.984	.982	.330	.721	.767	.743
Perceptron	+bigram	.023	.979	.984	.981	.316	.733	.772	.752
(BOTH)	tf-issf	.016	.987	.989	.988	.274	.778	.820	.798
	+bigram	**.014**	**.989**	**.990**	**.989**	.256	.793	**.827**	.810
Tokenizer	MeCab+IPADic	.155	.902	.865	.883	.424	.718	.624	.667
	MeCab+UniDic	.004*	.998*	.996*	.997*	.377	.704	.767	.734
	JUMAN	.105	.934	.908	.921	.282	**.818**	.751	.783
	Kytea	.010*	.992*	.993*	.993*	**.254**	.798	.823	**.811**
	RakutenMA	.077*	.936*	.953*	.944*	.383	.700	.752	.725

Table 2: Segmentation results for katakana words in BCCWJ and katakana Twitter hashtags. Following Hagiwara and Sekine (2013), Kytea, MeCab with UniDic (MeCab+UniDic), and RakutenMA results on BCCWJ are reported here for reference since these tokenizers use BCCWJ as a training corpus.

ceptron and tf-issf against that of state-of-the-art Japanese tokenizers JUMAN 7.01 (Kurohashi et al., 1994); MeCab 0.996 (Kudo et al., 2004) with two different dictionaries, IPADic (Asahara and Matsumoto, 2003) and UniDic (Den et al., 2007); Kytea 0.4.7 (Neubig et al., 2011); and RakutenMA (Hagiwara and Sekine, 2014).

3.3 Results

We use precision (P), recall (R), F1-score (F1), and word error rate (WER) to evaluate the performance of each method. The evaluation results are shown in Table 2.[8]

The use of tf-issf in the UNLABEL setting outperforms the other frequency-based method with statistical significance under McNemar's Test with $p < 0.01$ and yields comparable performance against supervised methods on BCCWJ. In Table 2, we show that tf-issf outperforms the frequency-based method proposed by Nakazawa et al. (2005). Although tf-issf only uses the statistics from Wikipedia, it achieves superior performance to MeCab with IPADic (MeCab+IPADic) and comparable performance to JUMAN.

The main limitation of using tf-issf alone is that it cannot completely avoid the frequency bias of the corpus. For instance, the most frequent katakana sequence occurring in Japanese Wikipedia is リンク *linku* ("link"), which is both ambiguous—potentially referring to either "rink" or "link"—and frequent, because it is the abbreviation for "hyperlinks". As a result, the tf-issf score of this string is much higher than average, which causes the word エナジードリンク *enajīdo**linku*** ("energy drink") to be segmented as エナジー / ド / リンク *enajī / do / linku* ("energy / d / rink"). This problem can be ameliorated by incorporating BCCWJ to readjust the tf-issf values.

Table 2 also shows the segmentation result for Twitter hashtags. Here, the tf-issf values are readjusted using the structured perceptron and the whole of the BCCWJ core data to make a fair comparison with other tokenizers. Incorporating tf-issf into the structured perceptron improves the F1-score, from .743 to .798, when combined with unigrams. Although Kytea performs slightly better in terms of F1-score, tf-issf combined with bigrams achieves slightly higher recall because of fewer OOV words.

Table 3 shows the examples of segmentations produced for the OOV words that are not present in the BCCWJ training data. Tokenizers trained on BCCWJ except for RakutenMA fail to segment スマホケース "smartphone case" because the word スマホ "smartphone" does not appear in BCCWJ. Using tf-issf alone is also not sufficient to produce correct segmentations for all examples, and only tf-issf combined with structure perceptron successfully segments all examples.

4 Related Work

We now review relevant work on Japanese segmentation and describe the key ways in which our approach differs from previous ones.

Japanese word segmentation has an extensive history, and many Japanese tokenizers have been developed, from the rule-based tokenizer JUMAN (Kurohashi et al., 1994) to statistical tokenizers, MeCab (Kudo et al., 2004), Kytea (Neubig et al., 2011), and RakutenMA (Hagiwara and Sekine, 2014). However, these Japanese tokenizers require either manual tuning or a manually labeled corpus.

[8] Because the length feature only degrades the segmentation performance, we exclude the results from the tables.

Method	“Smartphone Apps”	“Ueno Daiki”	“My Number”
Gold data	スマホ / アプリ *sumaho / apuri*	ウエノ / ダイキ *ueno / daiki*	マイ / ナンバー *mai / nanbā*
Baseline	スマホ / アプリ *sumaho / apuri*	ウ/エ/ノ/ダ/イ/キ *u/e/no/da/i/ki*	マイ / ナンバー *mai / nanbā*
tf-issf (UNLABEL)	スマホ / アプリ *sumaho / apuri*	ウエノ / ダイキ *ueno / daiki*	マイナンバー *mainanbā*
tf-issf (BOTH)	スマホ / アプリ *sumaho / apuri*	ウエノ / ダイキ *ueno / daiki*	マイ / ナンバー *mai / nanbā*
MeCab+IPADic	スマホアプリ *sumahoapuri*	ウエノ / ダイキ *ueno / daiki*	マイ / ナンバー *mai / nanbā*
MeCab+UniDic	スマホアプリ *sumahoapuri*	ウエノ / ダイキ *ueno / daiki*	マイ / ナンバー *mai / nanbā*
JUMAN	スマホ / アプリ *sumaho / apuri*	ウエノダイキ *uenodaiki*	マイ / ナンバー *mai / nanbā*
Kytea	スマホアプリ *sumahoapuri*	ウエノ / ダイキ *ueno / daiki*	マイ / ナンバー *mai / nanbā*
RakutenMA	スマホ / アプリ *sumaho / apuri*	ウエノ / ダイキ *ueno / daiki*	マイナンバー *mainanbā*

Table 3: Examples of segmentation results for katakana words in Twitter hashtags using different segmentation methods. The correct segmentations are produced by tf-issf (BOTH) on these examples, while all others fail to achieve this.

4.1 Approaches Using Unlabeled Corpora

Closer to our own work, Koehn and Knight (2003) and Nakazawa et al. (2005) investigate segmenting compound words using an unlabeled corpus. These approaches do not achieve high precision on katakana words (Kaji and Kitsuregawa, 2011), however. To improve the segmentation accuracy, Kaji and Kitsuregawa (2011) incorporate a rule-based paraphrase feature (e.g., a middle dot “・”) to use an unlabeled corpus as training data without manual annotation. This method still requires manual selection of the characters used as word boundaries. Other studies use transliterations to segment katakana words using explicit word boundaries from the original English words (Kaji and Kitsuregawa, 2011; Hagiwara and Sekine, 2013). However, as not all katakana words are transliterations, it is advantageous to use a monolingual corpus.

4.2 TF-IDF-based Segmentation

Some similar work has been done on Chinese. Xiao et al. (2002) used tf-idf of context words to resolve segmentation ambiguities of Chinese words, but this approach assumes only two segmentation forms: combined and separated. This is adequate for two-character words in Chinese, which comprise the majority of Chinese words (Suen, 1986), but not for potentially very long katakana words in Japanese. In contrast to their approach, we regard each katakana term as one document and compute the inverse document frequency. The tf-issf approach also does not require context words since we compute the term frequency of each katakana term in question instead of the frequency of its context words. Thus, we need not assume that the training corpus has been automatically segmented by an existing tokenizer, which might include segmentation errors involving context words.

In contrast to these approaches, we use a new frequency-based method, inspired by tf-idf that uses an unlabeled corpus to tackle word segmentation of character sequences of unbounded length.

5 Conclusion

In this paper, we introduce tf-issf, a simple and powerful word segmentation method for Japanese katakana words. We show that using tf-issf alone outperforms the baseline frequency-based method. Furthermore, when tf-issf is incorporated into the structured perceptron together with simple features on a manually labeled corpus, it achieves comparable performance to other state-of-the-art Japanese tokenizers, outperforming all in recall.

5.1 Future Work

While our work focuses on the peculiarities of Japanese katakana words, tf-issf may be applicable to other languages. We leave this for future work. Further research is also necessary to determine the extent to which tf-issf is dependent on the domain of the corpora, and how transferable these gains are across various domains. Investigating the phonetic and corresponding orthographic changes that occur with shortened Japanese katakana words and their transference to new compounds may also lead to further improvements in segmentation results.

Acknowledgments

We would like to thank the anonymous reviewers and Michael J. Paul for providing helpful comments.

References

Enrique Alfonseca, Slaven Bilac, and Stefan Pharies. 2008. Decompounding Query Keywords from Compounding Languages. In *Proceedings of ACL-HLT*, pages 253–256.

Masayuki Asahara and Yuji Matsumoto. 2003. Ipadic version 2.7.0 user's manual (in Japanese). *NAIST, Information Science Division.*

Piyush Bansal, Romil Bansal, and Vasudeva Varma. 2015. Towards Deep Semantic Analysis of Hashtags. In *Proceedings of ECIR*, pages 453–464.

Martin Braschler and Bärbel Ripplinger. 2004. How effective is stemming and decompounding for german text retrieval? *Information Retrieval*, 7(3-4):291–316.

James Breen. 2009. Identification of Neologisms in Japanese by Corpus Analysis. In *E-lexicography in the 21st century: New Challenges, New Applications*, pages 13–21.

Ruey-Cheng Chen. 2013. An Improved MDL-based Compression Algorithm for Unsupervised Word Segmentation. In *Proceedings of ACL*, pages 166–170.

Michael Collins. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of EMNLP*, pages 1–8.

Yasuharu Den, Toshinobu Ogiso, Hideki Ogura, Atsushi Yamada, Nobuaki Minematsu, Kiyotaka Uchimoto, and Hanae Koiso. 2007. The Development of an Electronic Dictionary for Morphological Analysis and its Application to Japanese Corpus Linguistics (in Japanese). *Japanese Linguistics*, 22:101–123.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual Dependencies in Unsupervised Word Segmentation. In *Proceedings of ACL*, pages 673–680.

Masato Hagiwara and Satoshi Sekine. 2013. Accurate Word Segmentation using Transliteration and Language Model Projection. In *Proceedings of ACL*, pages 183–189.

Masato Hagiwara and Satoshi Sekine. 2014. Lightweight Client-Side Chinese/Japanese Morphological Analyzer Based on Online Learning. In *Proceedings of COLING*, pages 39–43.

Nobuhiro Kaji and Masaru Kitsuregawa. 2011. Splitting Noun Compounds via Monolingual and Bilingual Paraphrasing: A Study on Japanese Katakana Words. In *Proceedings of EMNLP*, pages 959–969.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of EACL*, pages 187–193.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis . In *Proceedings of EMNLP*, pages 230–237.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese Morphological Analyzer JUMAN. In *Proceedings of The International Workshop on Sharable Natural Language*, pages 22–28.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written Japanese. *Language Resources and Evaluation*, 48(2):345–371.

Daichi Mochihashi, Takeshi Yamada, and Naonori Ueda. 2009. Bayesian Unsupervised Word Segmentation with Nested Pitman-Yor Language Modeling. In *Proceedings of ACL-IJCNLP*, pages 100–108.

Fred Morstatter, Jürgen Pfeffer, Huan Liu, and Kathleen M. Carley. 2013. Is the Sample Good Enough? Comparing Data from Twitter's Streaming API with Twitter's Firehose. In *Proceedings of ICWSM*, pages 400–408.

Toshiaki Nakazawa, Daisuke Kawahara, and Sadao Kurohashi. 2005. Automatic Acquisition of Basic Katakana Lexicon from a Given Corpus. In *Proceedings of IJCNLP*, pages 682–693.

Graham Neubig, Yosuke Nakata, and Shinsuke Mori. 2011. Pointwise Prediction for Robust, Adaptable Japanese Morphological Analysis. In *Proceedings of ACL-HLT*, pages 529–533.

Hideki Ogura, Hanae Koiso, Yumi Fujike, Sayaka Miyauchi, and Yutaka Hara. 2011. Morphological Information Guildeline for BCCWJ: Balanced Corpus of Contemporary Written Japanese, 4th Edition (in Japanese). Research report.

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*, pages 1773–1782.

Gerard Salton and Christopher Buckley. 1988. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523.

Ching Y Suen. 1986. *Computational Studies of the Most Frequent Chinese Words and Sounds*, volume 3. World Scientific.

Luo Xiao, Sun Maosong, and Tsou Benjamin. 2002. Covering Ambiguity Resolution in Chinese Word Segmentation Based on Contextual Information. In *Proceedings of COLING*, pages 1–7.