Sudachi: a Japanese Tokenizer for Business

Kazuma Takaoka[†], Sorami Hisamoto[†], Noriko Kawahara[†],
Miho Sakamoto[†], Yoshitaka Uchida[†], Yuji Matsumoto[‡]
[†]Works Applications
[‡]Nara Institute of Science and Technology
{takaoka_k hisamoto_s, kawahara_n, sakamoto_mi, uchida_yo}@worksap.co.jp, matsu@is.naist.jp

Abstract

Tokenization, or morphological analysis, is a fundamental and important technology for processing a Japanese text, especially for industrial applications. However, we often face many obstacles, such as the inconsistency of token unit in different resources, notation variations, discontinued maintenance of the resources, and various issues with the existing tokenizer implementations. In order to improve this situation, we develop a tokenizer called *Sudachi* and its accompanying dictionary with features such as multi-granular output and normalization of notation variations. In addition to this, we continuously maintain our software and language resources in long-term as a part of the company business. We release the resulting tokenizer software and language resources freely available to the public as an open source software. You can access them at https://github.com/WorksApplications/Sudachi.

Keywords: Tokenization, Morphological Analysis, Segmentation, Part-of-Speech Tagging, Lemmatization, Open Source Software

1. Introduction

Unlike whitespace separation between words for English text, Japanese text does not contain explicit word boundary information. We need unobvious methods to recognize words within a text. For Japanese text, the process equivalent to "tokenization" in other languages is often called "morphological analysis". It consists of three subprocesses, namely segmentation, part-of-speech (POS) tagging, and lemmatization (In the following we use the word "tokenization" to indicate these three processes).

The definition of a token in Japanese is not trivial; People develop various systems and resources, each with different kinds of the standard. Lack of token unit compatibility is one of the critical problems of Japanese language resources. When we look at the applications of Japanese text processing in various business scenes, applying parsing or more advanced language process is not common. It is typical to just conduct tokenization and use its post-processed output. For many companies tokenization is a fundamental and important technology for text processing. However, when increasing number of companies are demanding Japanese text processing recently, we are lacking freely available and useful resources for tokenization.

In order to improve this situation, we develop a new Japanese tokenizer and dictionary for business use. We make them available to the public as an open source software (OSS).

2. Previous Work

2.1. Japanese Tokenizers

When conducting Japanese tokenization for business applications, in the majority of cases *MeCab*[1] (Kudo et al., 2004) or *Kuromoji*[2] (the re-implementation of MeCab) are used.

MeCab can process text at excellent speed, however, its functions are limited to segmentation, POS tagging, and lemmatization; Users need to pre-/post-process the text by themselves. It is common to conduct text formatting, sentence segmentation, and character normalization as pre-processing. Typical post-processing includes simple chunking (e.g., for numeric expressions) and filtering by POS tags. Each user performs these processes on their own, therefore we tend to reinvent the wheel, or conduct such processes in inefficient ways.

There are two versions of Kuromoji, the standalone tool and the one integrated into a search library Apache Lucene[3]. The former version has issues similar to MeCab, and for the latter, although it provides some pre-/post-processing functions as part of the search system, we can not use them outside Lucene.

2.2. Language Resources

For the systems such as MeCab and Kuromoji, the language models are independent of the system, in form of dictionaries. The user may select a resource for tokenization from publicly available choices.

IPADIC (Asahara and Matsumoto, 2003) is the most widely used resource for Japanese tokenization; However it has not been updated since 15 years ago, therefore the dictionary lacks new words and the bug fixes have not been applied.

NAIST Japanese Dictionary[4], a dictionary developed based on IPADIC, aimed to solve the license issues of IPADIC, as those issues make it difficult to use the resource for OSS purposes. However it is currently not widely used, as the dictionary lacks some essential vocabularies, and IPADIC license issues have been solved subsequently.

UniDic[5] (Den et al., 2007; Kouno and Ogiso, 2015) (National Institute for Japanese Language and Linguistics, 2017) is a project to develop a Japanese electronic dictionary with uniformity and identity. The outcome is used

[1] http://taku910.github.io/mecab/
[2] https://www.atilika.com/ja/kuromoji/
[3] https://lucene.apache.org/
[4] https://ja.osdn.net/projects/naist-jdic/
[5] http://pj.ninjal.ac.jp/corpus_center/unidic/

for building Corpus of Spontaneous Japanese (CSJ)[6] and Balanced Corpus of Contemporary Written Japanese (BCCWJ)[7] (Maekawa, 2008). The project also offers a dictionary for conducting tokenization using MeCab. To emphasize the reproducibility of annotation, it adopts shorter token units in order for the annotators to process text unambiguously. To ensure the reproducibility the segmentation rules are defined as operating procedures, therefore it may get annotated in unintuitive fashions. It is shown effective for search purposes (Takahashi and Sassano, 2016), however, it is not suitable for syntactic or semantic analysis (National Institute for Japanese Language and Linguistics, 2017). The number of language resources derived from BCCWJ is growing, therefore UniDic is adopted more in the academic fields.

NEologd[8] (Sato et al., 2016; Sato et al., 2017) is a resource released as an add-on for other dictionaries such as IPADIC or UniDic. It consists of new words that are not included in those dictionaries. These new words are extracted from many language resources on the Web automatically or semi-automatically, and it is frequently updated (currently twice a week). This dictionary contains vocabularies in longer unit compare to other resources; We may say that these vocabularies are more of named entities (NEs) instead of words. If we apply these vocabularies directly to the search systems, we tend to get missing search results, as the shorter tokens that constitute the NEs are not indexed. The authors claim that the user can get shorter tokens by recursively tokenizing these NEs. The recursive tokenization can be erroneous, however, the authors do not discuss the negative impact thereof.

3. Japanese Tokenizer for Business

Given the situation we described in the previous section, we aim to develop a tokenizer and dictionary for business use. In more detail, by "for business use" we mean the following:

- Available for a variety of applications and purposes
- Sufficient accuracy
- Sufficient coverage
- Long-term continuous maintenance

We release our software and language resources freely available to the public (Works Applications, 2018).

Table 1 shows the comparisons between the language resources for Japanese tokenization.

3.1. Multi-granular Tokenization

How to define the granularity of the token unit in Japanese tokenization has long been discussed. However, the suitable unit differs for each application. For example, the search systems need shorter units in order to ensure high recall rate, whereas we want longer units for semantic analysis in order to recognize the entities. Hence we manually annotate the constituting shorter units within the longer units, in order to provide tokens of different granularity for each application purpose.

We define three types of units:

- **Short**: Compatible with UniDic
- **Middle**: Similar to the "words" in general sense
- **NE**: Named entities

For each unit, we manually annotate its word structure constituted by the shorter units.

To be precise, Named Entity Recognition (NER), detecting NEs in a text, is not a subject of tokenization. However to conduct accurate NER we need some kind of lexical information, and it is more efficient to register these NEs to the dictionary and process them together at the tokenization step.

Figure 1 shows an example of multi-granular tokenization.

Figure 1: An example of multi-granular tokenization. For short unit the input is tokenized into the short parts, whereas for NE unit the result refers to the name of an existing museum.

3.2. Normalization

The Japanese language has a complicated written form system, and it does not have a rigorous orthography. This makes the notation variation (表記揺れ) a severe problem for processing Japanese text. It is essential to solve this issue in order to process text available in the real world. Table 2 shows different types of notation variation.

To deal with this problem, we manually add normalized forms for the vocabularies in our dictionary.

3.3. Continuous Maintenance

For the languages, new words appear as time passes, and the usage of existing words may change as well. Therefore it is essential to maintain the dictionary continuously.

In Japan, the maintenance of the resources developed by national universities and institutes is often discontinued after some time. On the other hand, we continuously maintain our resources in long-term as a part of the company business. NEologd also claims to continue the maintenance, however, it is created automatically or semi-automatically, therefore the part of the resource have poor quality, and it cannot deal with normalization of notation variations.

3.4. Software for Tokenization

We release a tokenizer software called *Sudachi*[9] as an OSS in order to use the language resources we have developed.

[6] http://pj.ninjal.ac.jp/corpus_center/csj/

[7] http://pj.ninjal.ac.jp/corpus_center/bccwj/

[8] https://github.com/neologd

[9] https://github.com/WorksApplications/Sudachi

	IPADIC	UniDic	NEologd	Sudachi
Multi-granularity	No	No	No	**Yes**
Named Entity	Some	Some	**Yes**	**Yes**
Normalized Form	**Yes**	**Yes**	No	**Yes**
Continuous Maintenance	No	No	**Yes**	**Yes**
Manual Check	**Yes**	**Yes**	No	**Yes**

Table 1: Comparisons between the resources for Japanese tokenization.

Type	Example
Kana Suffix Variation (送り違い)	打込む / 打ち込む
Character Type Variation (字種)	かつ丼 / カツ丼
Glyph Variation (異体字)	附属 / 付属
Misspelling (表記誤り)	シミュレーション / シュミレーション

Table 2: Types of notation variation in Japanese. Words with different notations, indicating the same meaning.

This tokenizer has functions such as tokenization in different granularity, and the normalization of notation variations.

We also design the pre-/post-processes to be plugins of the tokenizer, aiming to aggregate the knowledge of various users that were previously scattered in individual user's environment. We implement the plugins and release them to the public, so that anyone can easily conduct Japanese tokenization without having a detailed knowledge of the task.

The original version is implemented in Java. We also release the Python version called *SudachiPy*[10]. In addition to the tokenizer itself, we also develop and release a plugin[11] for Elasticsearch[12], an open source search engine.

4. Current Status

As we described in subsection 2.2., the number of language resources derived from BCCWJ is growing; Therefore we emphasize the compatibility to UniDic in order to make use of those resources. We hence develop our resources based on UniDic's tokenization dictionary, add middle and NE unit vocabularies from NEologd and other resources, then adjust the word structures and normalized forms.

4.1. Short Unit: Revising UniDic

UniDic is designed to emphasize the reproducibility of annotation, to have segmentation in a good order. This makes the annotation consistent regardless of who conducted it, however, it sometimes has unintuitive segmentation and this causes issues for practical usages.

For example, UniDic has the rule to decide segmentation according to its origin (語種) as Japanese, Chinese or Western; For Japanese origin words, it considers the word base and the suffix together as a unit, whereas for Chinese origin words they are segmented into 2 units. Table 3 shows a segmentation example where the behaviors are different even with the same suffix.

We manually modify these unintuitive segmentations for the practical purposes.

[10]https://github.com/WorksApplications/SudachiPy

[11]https://github.com/WorksApplications/elasticsearch-sudachi

[12]https://www.elastic.co/jp/products/elasticsearch

UniDic Segment	Origin (語種)
使用 / 料 (Charge)	Chinese (漢語)
為替料 (Exchange Fee)	Japanese (和語)

Table 3: An example of segmentation difference between the tokens with the same suffix in UniDic. They have the same suffix "料", however, the resulting segments are different becasuse of their origins.

4.2. NE Unit: Revising NEologd

The words in NEologd are collected automatically or semi-automatically. Therefore we screen the resource to exclude entities that are included by unclear reason, or unnecessary for our purposes. For example, the date expressions are sometimes included in NEologd, however, we would like to exclude and handle them in pre-/post-processing steps instead. We then manually add the word structure and normalized form. We also fix *kana* information if necessary, as it is automatically estimated in NEologd and it occasionally is inaccurate.

4.3. Middle Unit: Selection Policies

Middle unit is the unit that is most close to the "words" the users would expect, and it will be useful for different purposes. However, it is difficult to comprehensively define this unit.

Currently, we are investigating the definition for each category separately. If the token is a noun, we define the unit to include its prefix and suffix. On the other hand, if it is a verb we include up to the compound verb to be a middle unit.

4.4. Vocabulary Size

We currently selected 2.6 million tokens, and 1.4 millions of them are accurately given normalized forms, POS, and *kana* information. Among them, 0.8 million tokens have word structure information.

5. Future Work

The language of Computer-mediated Communication (CMC) such as in e-mails, blogs, and social networks is called Uchi-kotoba (打ち言葉, typing language). It has different characteristics compared to written and spoken languages. Uchi-kotoba has unique spellings, simplifications,

and phase vocabularies. We expect to have an even larger amount of such text in the future, and the emphasis on processing such text will increase.

It is difficult to minimize the negative effect when incorporating the NEs that may result in erroneous analysis. Transient NEs such as the titles of the movies, TV shows or the names of the political parties[13] may become the cause of erroneous analysis. We need to devise methods in order to minimize such errors, for example, by removing them after a certain period of time when the frequency of appearance has decreased.

Documents handled in the business scenes may contain noises. These noises may result from typos while inputting text, or failures of speech recognition or optical character recognition. We would like to achieve a robust tokenization that minimizes the negative impact to the surroundings of these noise sections. It is also important to develop a tokenizer with error correction features for such noises.

There are situations which we have not yet decided how to handle in our formulation. One situation is when a word has ambiguous word structure. For example, a word "暴力団員" (Yakuza, a gang member) can be formed of either "暴力 / 団員" (violence / organization member) or "暴力団 / 員" (gang / member). We want both structures in applications. Another situation is when it is not obvious how to annotate a word. For example, the compound of "輸出" (export) and "輸入" (import) is "輸出入" (export and import), where the common prefix "輸" is shared and appears only once.

6. Conclusion

In this paper, we presented a Japanese tokenizer Sudachi and its dictionary, aiming to improve the current situation of the Japanese tokenization task especially for the business application purposes.

We first described the importance of tokenization for Japanese text processing in business applications.

We then showed the problems of the current tokenization tools and language resources. Pre-/post-processing are the important parts of text processing, however, the existing tokenizers do not have these processes built into the tool, or not easily usable for the users' needs. This tends to force the users to reinvent the wheel or conduct such processes in inefficient ways. Existing language resources for Japanese tokenization are often not maintained continuously, or the token unit of the resource may not be suitable for the users' purposes.

We aim to solve such problems by developing a high-quality dictionary with multi-granular token information for different purposes, normalized form information for notation variations, long-term continuous maintenance as a part of the company business, and an accompanying tokenizer with pre-/post-processes as plugins. We make our tokenizer and language resources freely available to the public as an OSS.

We explained the current status and issues of the project.

Lastly, we mentioned various topics that need to be dealt to improve tokenization quality.

[13]Political parties in Japan are often formed for a short-term, and they are occasionally titled using common names.

7. Acknowledgments

We thank Dr. Masayuki Asahara of National Institute for Japanese Language and Linguistics and Toshinori Sato of LINE Corporation for the discussions. We also thank the Sudachi users and the anonymous reviewers for the comments.

8. Bibliographical References

Asahara, M. and Matsumoto, Y., (2003). *ipadic version 2.7.0 ユーザーズマニュアル (IPADIC version 2.7.0 Users Manual, in Japanese)*. Nara Institute of Science and Technology.

Den, Y., Ogiso, T., and Ogura, H. (2007). The development of an electronic dictionary for morphological analysis and its application to japanese corpus linguistics (in japanese). *Japanese linguistics*, 22:101–123.

Kouno, T. and Ogiso, T. (2015). Improving an electronic dictionary for morphological analysis of japanese: Use of historical period information. In *Proc. of The 9th International Conference of Asian Association of Lexicography (ASIALEX)*, pages 441–449.

Kudo, T., Yamamoto, K., and Matsumoto, Y. (2004). Applying conditional random fields to japanese morphological analysis. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 230–237.

Maekawa, K. (2008). Balanced corpus of contemporary written japanese. In *Proc. of The 6th Workshop on Asian Language Resources (ALR), The 3rd International Joint Conference on Natural Language Processing (IJCNLP)*, pages 101–102.

National Institute for Japanese Language and Linguistics, (2017). *UniDic とは? (What is UniDic?, in Japanese)*. `http://unidic.ninjal.ac.jp/about_unidic`.

Sato, T., Hashimoto, T., and Okumura, M. (2016). Operation of a word segmentation dictionary generation system called neologd (in japanese). In *Information Processing Society of Japan, Special Interest Group on Natural Language Processing (IPSJ-SIGNL)*, pages NL–229–15.

Sato, T., Hashimoto, T., and Okumura, M. (2017). Implementation of a word segmentation dictionary called mecab-ipadic-neologd and study on how to use it effectively for information retrieval (in japanese). In *Proc. of the Twenty-three Annual Meeting of the Association for Natural Language Processing*, pages NLP2017–B6–1.

Takahashi, F. and Sassano, M. (2016). 情報検索のための単語分割一貫性の定量的評価 (quantitative evaluation of word segmentation consistency for information retrieval, in japanese). In *Proc. of the Twenty-two Annual Meeting of the Association for Natural Language Processing*, pages 949–952.

9. Language Resource References

National Institute for Japanese Language and Linguistics. (2017). *UniDic*. 2.1.2, ISLRN 114-759-406-461-7.

Works Applications. (2018). *Sudachi Dictionary*. 0.1.0, ISLRN 811-243-395-600-3.