Topics in Natural Language Processing

Japanese Morphological Analysis

Shijie Yao s1770642

March 2018

Contents

Abstract

This paper discusses how traditional mainstream methods and neural-network-based methods have been applied to the task of Japanese morphological analysis in Natural Language Processing (NLP). The traditional methods, including rules, Hidden Markov Model (HMM), Conditional Random Fields (CRF), and Support Vector Machine (SVM), are mostly proposed more than 10 years ago. Based on these methods, a couple of morphological analysers are built, such as JUMAN, ChaSen, MeCab, and KyTea. Until recently, applying neural methods to Japanese morphological analysis is rarely seen. This paper introduces the various methods that have been proposed, collects information of Japanese corpora and dictionaries for NLP research, evaluates several morphological analysers on Japanese lemmatisation task, and proposes future directions based on recurrent neural networks language modelling.

Keywords: morphological analysis, Japanese morphological analysis, Japanese natural language processing

1 Introduction

1.1 Morphological analysis

Morphology is the subject in linguistics which studies word formation. Morphological analysis is one of the most important upstream tasks in Natural Language Processing (NLP). It is useful for a lot of downstream NLP tasks such as information extraction or web search. For example, morphological analysis can return the morphological variants for the words in users' query and enlarge the search scope, thus being able to find more related web pages. Generally, morphological analysis is to annotate natural language with morphological information. Such morphological annotation might include but not limited to: part-of-speech (POS) tags, morpheme annotations, lemmas, etc. For different languages however, the specific tasks in morphological analysis might vary a lot. For languages such as English where words are naturally separated by white spaces, there is no need to do word segmentation. But for languages such as Japanese, Chinese and Thai, word segmentation is the very first step of morphological analysis. As a pre-requisite, these languages need to be segmented into tokens such as words or morphemes delimited by white spaces.

Research in morphological analysis can benefit not only NLP scholars but also corpus linguists. Morphological analysis helps with the construction of corpora, which are then can be used as training data to be learnt by morphological analysers or test data to test their performance. This reciprocity relationship makes morphological analysis and corpora construction sounds more or less a chicken-and-egg problem.

1.2 Japanese morphological analysis

Japanese is a special language in that words are not separated by white spaces and the writing system is very complicated. The three main set of writing symbols in Japanese are: *kanji* (漢字 essentially Chinese characters), *hiragana* (平仮名 Japanese character symbols representing original Japanese words), and *katakana* (片仮名 Japanese character symbols mainly representing load words). Even for words written in different writing symbols, there is no explicit white spaces. Thus it is very likely to cause ambiguity in Japanese. For example, in the sentence この先生きのこれるのか, the most likely way to segment it will be: この先/生きのこれる/のか which means *Can I survive from now on?*. But it could cause ambiguity as it is also possible to be segmented as この先生/き/のこれる/のか which means *Can the teacher still be alive?*. This phenomenon is called ambiguous word boundaries, one of the major problems and challenges in Japanese morphological analysis [14], as well as the first task of it. In actual, for most of the Japanese morphological analysers and annotated corpora which serve as golden standard, we can tell that Japanese sequences are normally segmented at morpheme-like (文節 *bunsetsu*[1]) level, instead of word level. This has become the accepted criterion of Japanese *word* segmentation (which is in actual *bunsetsu* segmentation).

The second task for Japanese morphological analysis is POS tagging as soon as the sentences are segmented into *bunsetsu* (hereinafter, for simplicity we will still use *word segmentation*). However, so far, as there has not formed any consensus on a fixed set of POS tags for Japanese, the annotation of POS tags might vary between corpora. Some common and recognised POS types in Japanese are: verbs (*godan*-verbs, *ichidan*-verbs and irregular verbs), adjectives (*i*-adjectives and *na*-adjectives), nouns, adverbs, conjunctions, auxiliary verbs, etc. Among them, verbs, adjectives and auxiliary verbs can be inflected (活用 *katsuyou*). Inflection does not change the POS of the word but might change its

[1] the minimal unit that is reasonable to be segmented, which is different from 単語 *word*

meaning. Some examples of words and one of their inflected forms (here inflection of informal forms to the corresponding formal forms) are shown in Table 1. Overall, Japanese morphological inflection tends to follow some phonological pattern (see the example of 行く in the next paragraph), and most can be derived by looking up in a dictionary. So it was even feasible to make a grammar to cover almost all the inflection rules [1]. That is why some early rule-based (details in 2.1) morphological analysers could work so well in lemmatisation, which is another task in Japanese morphological analysis, following its two pre-requisites, word segmentation and POS tagging.

Lemmatisation takes as input an inflected form and outputs its lemma which is a dictionary entry. Take some English words as instances, the lemma for the word *going* is *go* and for *went* is *go* as well. But the former one is regular, as appending *ing* to most English verbs converts them into progressive tense; while the latter is irregular, as for most English verbs, appending *d* or *ed* derives the verbs into their past tense forms, whilst *went* does not follow the rule. In the case of Japanese inflection, there are regular and irregular cases as well. For example, the inflection of 行く *i* ***ku*** → 行きます *i* ***ki*** *ma su* is viewed as regular because the regular inflection rule is to change the vowel ***u*** in *ku* into ***i*** resulting *ki*. In contrast, the inflection of ござる *go za* **ru** → ございます *go za* **i** *ma su* is viewed as irregular, otherwise the inflection, if obeying the phonological rule, should be *go za* ***ri*** *ma su* which keeps the consonant *r* and only changes the vowel from ***u*** into ***i***.

	Verb			**Adjective**		**Auxiliary verb**
	godan	***ichidan***	**irregular**	***i***	***na***	
Basic form	走る	寝る	する	良い	綺麗	れる
Inflected form	走った	寝た	した	良かった	綺麗だった	れた

Table 1: POS types in Japanese that can be inflected: examples of words and one of their inflected forms

2 Japanese morphological analysis based on traditional mainstream methods

Since 1992, various methods of doing Japanese morphological analysis and corresponding systems have been proposed and published. By *traditional methods*, we refer to methods of machine learning excluding deep learning such as artificial neural networks. We introduce different machine learning algorithms and the systems built on them one by one. There might be some overlapping of methods between different systems, but we will prioritise the core and novel part of each system that distinguishes it from the others.

2.1 Rule-based methods: JUMAN (from 1992)

Rule-based morphological analysers require extensive human involvement. In 1992, the first rule-based Japanese morphological analyser JUMAN was released [15, 16].

JUMAN recognises possible combinations of characters which can be looked up in the dictionary, and seeks for possible connections between neighbouring words using the connectivity dictionary which is essentially bigram information [15]. Bigram can represent the connectivity between two words because it catches the information of their co-occurrences. However, there are some cases where multiple word segmentations seem reasonable and meanwhile meet certain constraints of connectivity.

For example, the phrase 外国人参政権 [25] allows at least two ways of segmentation. Two of them are displayed in Table 2. Although both segmentations do not disobey the connectivity constraints on pairs of POS taggers (i.e. here they both follow the rule that nouns can follow nouns), the second segmentation 外国/人参/政権, which literally means *foreign countries/carrot/political power*, semantically does not make any sense at all.

Segmentation	POS tags	Literal meaning
外国人/参政権	noun/noun	foreigner/rights to participate in political affairs
外国/人参/政権	noun/noun/noun	foreign countries/carrot/political power

Table 2: Example of ambiguous word boundaries in Japanese that traditional Japanese morphological analysers are unable to handle

To deal with the problem of unreasonable word segmentation, in JUMAN, costs, are assigned to words and pairs of POS tags of neighbouring words [16]. The costs of words are determined by the frequency of each word itself, i.e. unigram frequency; the costs of POS tag pairs come from the frequency of POS tag bigrams. The rule is, the higher the frequency is, the lower the cost would be. Words and connectivities with higher costs are less likely to be a chunk. After assigning costs to the segmentation candidates, JUMAN constructs a lattice-like route map (see Figure 1 as an example), to sort out the segmentation of a given sentence with the lowest costs and outputs it (see Figure 2 as an example).

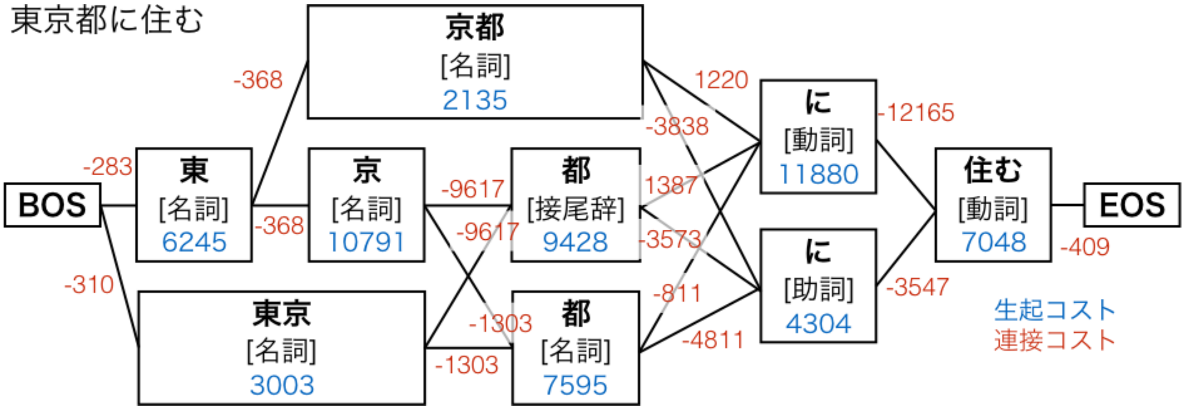

Figure 1: Lattice-like route map for the sentence 東京都に住む: BOS and EOS mean the beginning and the end of the sentence respectively, characters in bold constitute the sentence, words in square brackets indicate POS, blue numbers in the blocks are the costs of words, and red numbers on the edges are the costs of connectivity

However, there still remain some problems. Takeuchi [33] stated the problem of different costs of segmentation in different domains of text. This makes a morphological analyser hard to scale to wider domains, especially the web where the domain is usually unpredictable. Besides, as pointed out by [15], when the desired output is not assigned the highest cost, some manual correction has to be carried out. But any correction on one pair of words can potentially influence some other pairs, which can usually cause more incorrect outputs. Based on the error analysis, [15] found that fixing a couple

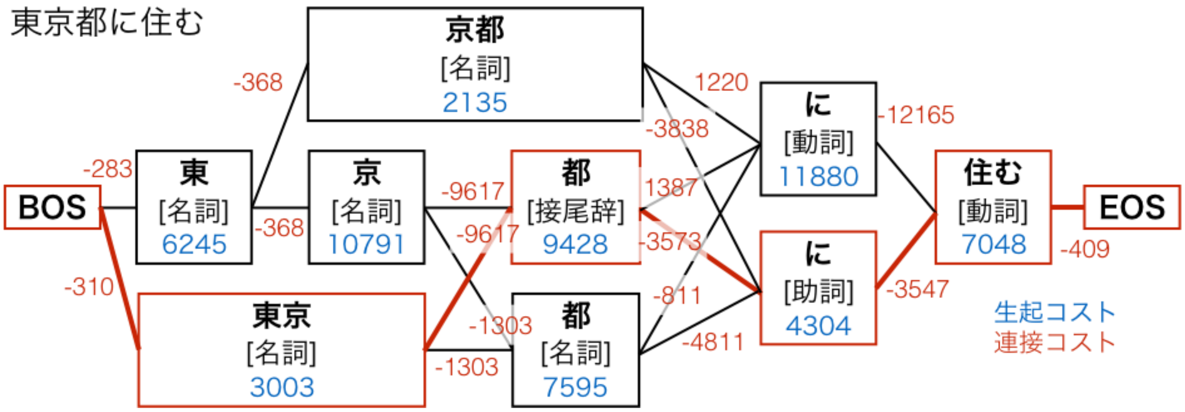

Figure 2: Lattice-like route map for the sentence 東京都に住む: BOS and EOS mean the beginning and the end of the sentence respectively, characters in bold constitute the sentence, words in square brackets indicate POS, blue numbers in the blocks are the costs of words, and red numbers on the edges are the costs of connectivity; the red edges draw the best path of the lowest cost

of fixed expressions composed of Japanese *kana* can solve most of the existing problems. By 1998, the accuracy of JUMAN in Japanese word segmentation is reported 99.0% by [15] but without specifying the test data and domain, which makes the comparison of performance hard. And the accuracy of JUMAN in word segmentation and POS tagging when tested on newspaper editorial articles is reported 93%∼95%.

Since its debut, JUMAN has undergone a series of improvement to be equipped with more sophisticated algorithms and be able to handle more unknown words and scale to the size and diversity of the web. Recently, the latest version of JUMAN (JUMAN 7.0)[2] integrates a dictionary containing words extracted from Wikipedia, which is updated periodically.

2.2 Hidden Markov Models: ChaSen (from 1996)

ChaSen[3] is a Japanese morphological analyser developed from its precursor JUMAN. Adapted from JUMAN, ChaSen defines costs of units and the connectivity of neighbouring units based on bi-gram as well. From this point of view, ChaSen is partly a rule-based morphological analyser as well. Then, based on the costs, both systems can find a most probable output with the lowest cost. But unlike JUMAN, according to [19], in ChaSen, costs are assigned to *morphemes* and the connectivity of two morphemes. The costs are directly learned from a POS-tagged Japanese newspaper article corpus. For unseen words, they are assigned a single POS tag for unknown words and a very high cost. For unseen bi-grams of morphemes, they are not allowed thus restricted; but users can allow them by giving them extremely high costs to make the morphological analysis not crash. To tune the costs, ChaSen makes use of a POS bi-gram Hidden Markov Model (HMM). HMM is essentially a generative finite state automaton (FSA) which maps a sequence of input to a sequence of states. In POS tagging, every state is tied with a POS tag. HMM has two types of probabilities as the parameters: emission probabilities and transition probabilities. Emission probabilities predict how likely the input is given

[2] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN JUMAN 7.0 retrieval date: 25/03/2018

[3] http://chasen-legacy.osdn.jp ChaSen retrieval date: 28/03/2018

the current POS tag; transition probabilities predict the probability of moving to the next state of POS tag given the current POS tag. The parameters can be learnt by the forward algorithm which is supervised learning, if the model is fed with pairs of a input sequence and a POS tag sequence; or by the forward-backward algorithm which is an unsupervised way that does not require tagged sequences as training data.

However, ChaSen is poor at lemma identification, and due to the inherent property of HMM, it is hard to introduce new features which could be potentially helpful, such as *goshu* [7] (more details in 2.3). This is because adding any new feature means adding to HMM new tags combining POS and the new feature, which could result in data sparsity. Besides, ChaSen also faces the problem of introducing negative effects as the costs are tunable by human hands as JUMAN. Furthermore, since both ChaSen and its precursor JUMAN heavily rely on lexicon, there is some research that attempted to abandon lexicon and instead only use counts of character bi-grams from a corpus to segment Japanese, which performed on par with or slightly better than ChaSen and JUMAN [2].

2.3 Conditional Random Fields: MeCab (from 2006)

Conditional Random Fields (CRF) are undirected graphic models which build probabilistic models to segment and label sequence data [17]. The objective of CRF is to maximise the conditional probability of the output sequence Y given the complete input sequence X. CRF is learned from pairs of a word and a label, i.e. $\langle w, t \rangle$, from the input and the output, assuming both sequences are of the same length. CRF has been proved to be successful in a wide range of NLP tasks [17] [21] [28] [32]. Apart from its application to Chinese word segmentation [28], Kudo [14] applied CRF to do Japanese word segmentation as well. We then explain CRF in the context of Japanese morphological analysis. Because the problem of ambiguous word boundaries in Japanese, the output size would not be fixed through all candidates. Taking that into consideration, Kudo [14] made some modification to the original formulation of CRF, by introducing the global feature vector for the output sequence. The global feature vector $\mathbf{F}(Y, X)$ is a concatenation of the following: $F_1(Y, X), ..., F_K(Y, X)$, where each $F_k(Y, X) = \Sigma_{i=1}^{length(Y)} f_k(\langle w_{i-1}, t_{i-1} \rangle, \langle w_i, t_i \rangle)$. Then the conditional probability $P(Y|X)$ can be written as $\frac{1}{Z_X} exp(\mathbf{W} \cdot \mathbf{F}(Y, X))$, where $\mathbf{W}$ is a whole matrix of learned weights or parameters associated with every feature functions f_k. Finally, simply by running the Viterbi algorithm, the most probable sequence $\hat{Y}$ given the input sequence X can be found by Equation (1) (c is a candidate from the candidate set C):

$$\hat{Y} = argmax_{c \in C} P(Y|X) = argmax_{c \in C} \mathbf{W} \cdot \mathbf{F}(Y, X) \tag{1}$$

One of the biggest advantages of discriminative CRF over generative HMM is that it can capture correlated features such as overlapping and hierarchical features, because CRF does not need to make any independence assumptions of features. CRF allows using correlated features at the same time. For example, Kudo [14] integrated four-level hierarchical POS tags provided by IPA[4] tagset mainly used in ChaSen. This tackles the problem of data sparsity if only using the bottom level POS tags, as it does more fine-grained categorisation; it also helps solve the lack of granularity if only using the top level POS tags, which provides rather broad categorisation. Besides, CRF is superior to HMM in that it also overcomes some other problems such label bias and length bias, which are the bottlenecks of HMM.

Based on the work of [14], another Japanese morphological analyser MeCab[5] was trained on the RWCP Text Corpus with ChaSen's standard lexicon. And MeCab outperformed ChaSen in both word

[4] http://chasen.naist.jp/stable/ipadic/ IPADIC dictionary used by ChaSen retrieval date: 25/03/2018

[5] http://taku910.github.io/mecab/ MeCab retrieval date: 25/03/2018

segmentation and POS tagging [7]. Kudo [14] further suggested adding more features such as n-gram information, which would potentially help improve the performance. However, this would lead to a trade-off between the quantity of features and the complexity of computation. So rather than simply increasing the number of features, whether and how the additional features could be helpful should be fully considered. On top of the work done by [14], Den [7] proposed to add *goshu* information as features to CRF for Japanese lemmatisation. *Goshu* classifies words by their origins, thus can be helpful disambiguate heteronyms, i.e. words sharing the same surface form but derived from different lemmas. *Goshu* information can be obtained from the UniDic dictionary. The use of *goshu* was justified by it improving the morphological analysers' performance on Japanese lemmatisation task.

2.4 Support Vector Machine: KyTea (from 2009)

Methods so far such as HMM [19] [20] and CRF [14] carried out Japanese morphological analysis based on the information of neighbouring words and POS tags. Unlike the above methods which learn from structured information, Neubig [27] proposed pointwise prediction: at each point making a single decision which is independent from the others while ignoring structured information.

Traditionally, joint sequence-based models have been used for morphological analysis, looking for a POS tag sequence with maximal likelihood given the input [14] [19] [20] [33]. Both the segmentation and the POS tagging task are viewed as classification problems: the segmentation task is a binary classification where only two situations (should be segmented & should not be segmented) exist; the POS tagging task is a multi-class classification problem where there is a set of tags to be assigned. Inheriting the classification idea from previous studies, Neubig [27] also treated them as classification problems, and applied Support Vector Machine (SVM) which is a pointwise classifier. The features used by [27] were those that can be directly calculated from the original string, e.g. the surrounding characters, the presence of words in the dictionary, etc., without any estimated annotation of word boundaries or POS tags.

SVM was proved to be excellent in not only binary classification but multi-class classification (can be treated as One-versus-Rest [26] binary classification). KyTea[6] is a Japanese morphological analyser which is a SVM classifier in essence. KyTea performs well at recognising unknown words and adapting different domains.

2.5 Comparison of traditional Japanese morphological analysers

Table 3 below compares the above-mentioned mainstream traditional Japanese morphological analysers. Part of Table 3 is adapted from the comparison table on the MeCab website[7]. Information that we are uncertain about is either marked ? or left blank.

From Table 3, we can clearly see the main changes in the development trend of Japanese morphological analysers happen in the algorithms being applied. The construction of rules might be hard for some morphologically-rich languages such as Turkish, Arabic, etc., but for Japanese it is rather easy due to its regularity in inflection as explained in 1.2. Almost all the analysers support all three tasks in Japanese morphological analysis except KyTea which does not support lemmatisation. Enabling users to use the systems to train models on their own corpora make these systems more flexible and comparable.

[6] http://www.phontron.com/kytea/ KyTea retrieval date: 28/03/2018

[7] http://taku910.github.io/mecab/#news the comparison table on MeCab website retrieval date: 28/03/2018

	JUMAN	**ChaSen**	**MeCab**	**KyTea**
Year of release	1992	1996	2002	2009
Analysis model	bi-gram	n-gram	bi-gram	n-gram
Cost determination	manual	corpus	corpus	N/A
Learning model	Rules	HMM	CRF	SVM
Corpora trained on	RWCP Text Corpus	RWCP Text Corpus	Kyoto University Corpus & RWCP Text Corpus	Balanced Corpus of Contemporary Written Japanese
N-best	No	No	Yes	Yes
Word segmentation	Yes	Yes	Yes	Yes
POS tagging	Yes	Yes	Yes	Yes
Lemmatisation	Yes	Yes	Yes	No
User training	Yes?	No?	Yes	Yes

Table 3: Comparison of four traditional Japanese morphological analysers: JUMAN, ChaSen, MeCab and Kytea

Table 3 might be far from complete regarding the number of aspects we focus on and the number of systems we select. Here we only list the aspects we think are most important for users when choosing systems that best suit their objectives. And we do not intend to cover all the Japanese morphological analysers so far being developed. There are some other analysers such as JTAG [8], which went one more step beyond JUMAN and ChaSen in that it makes use of word co-occurrence information to better select candidates when segmenting Japanese. But as it provided only a limited aspect of improvement and there is no usable system built based on it, JTAG is not described in details. Similar to the JTAG paper [8], there are some other papers [31] more or less attempting to improve Japanese morphological analysis by raising a single point and achieved visible improvements. But considering their influences, we did not include them.

3 Japanese morphological analysis based on neural methods

Traditional methods have dominated the field of Japanese morphological analysis for more than two decades, and they often perform at a satisfactory level. However, they sometimes still cannot handle semantic analysis [25] as shown in the example of 外国人参政権 (foreigners' right to participate in political affairs) in 2.1. In traditional n-gram models, there is a trade-off between the length of context to capture and data sparsity. For example, compared to bi-gram, tri-gram can include longer context, but meanwhile the data for training is sparser as there would be more unique tri-gram types. To tackle the context problem, recurrent neural network language model (RNNLM) [24] was proposed. RNNLM is a sequence-to-sequence language model which takes as input a sequence and outputs the probability of the next word given the embedding representation of the last word and the context. RNNLM is able to handle sequences of arbitrary lengths, thus potentially can capture as much context as possible.

Morita et al. [25] were the first to implement an RNNLM-based Japanese morphological analyser, i.e. JUMAN++[8]. They adapted the RNNME language model proposed by Mikolov [22] [23] which directly connects the input layer to the output layer, and acts as a maximum entropy model so that avoids wasting parameters for describing simple patterns [25]. As the training of neural networks normally require a large amount of data, Morita et al. [25] first trained the model using a raw web corpus of 10 million sentences [11] automatically pre-processed by JUMAN, and then re-trained the model on smaller whilst fine-grained hand-annotated corpora: the Kyoto University Text Corpus [12] and Kyoto University Web Document Leads Corpus [9]. Their RNNLM-based Japanese morphological analyser outperformed the four baselines including two traditional analysers JUMAN and MeCab, in both segmentation and POS tagging task, for both in-domain and out-of-domain. JUMAN++ is currently (at the point of March 2018) the latest and the only Japanese morphological analyser which integrates RNNLM (see Table 4 for comparison including JUMAN++).

Another problem that most of the traditional machine learning methods would encounter, as mentioned above, is data sparsity. Using RNN does not necessarily guarantee that the data sparsity problem is solved. For languages having a huge vocabulary such as English and Japanese, there would hardly ever be enough training data of every word type for training neural networks. To ensure enough training data for each *unit* type, various sub-word units thus have been proposed (mostly in Neural Machine Translation): Byte Pair Encoding (BPE) unit [29], character-level unit [13] [34] [35], syllable-level unit [4], morpheme-level unit [18], stroke-level unit in Chinese [30], etc. For some languages such as English, the size of a finite character-level dictionary (normally several hundred characters even including punctuations) is much smaller than that of a word-level dictionary (more than 170,000 words in the Second Edition of the Oxford English Dictionary) which is actually infinite as new words are being created at almost a daily basis. But for some other languages such as Japanese or Chinese, the size of a character-level dictionary might not necessarily be smaller than a word-level dictionary. Although people seem to be excited about trying out different sub-word units on various languages, it is still worth noting that some might not work as well as on some other languages. This indicates the need to take the specificity of languages into consideration.

Among all of the sub-word units above, the one that makes most sense in our opinions should be morphemes. Morphemes represent semantic meanings as they are the smallest meaningful units; meanwhile they capture syntactic relationships as well, such as agreement of gender, number, tense, etc. In the RNNLM-based Japanese morphological analysis paper [25], the training data were automatically segmented into morphemes by JUMAN. On one hand, this might be due to some criteria of corpus annotation or consensus among corpus linguistics and scholars of the Japanese language; on the other hand, this triggers our motivation to try out different types of units to observe how they would influence the performance of the RNNLM-based morphological analyser.

4 Corpora and dictionaries for Japanese morphological analysis

Corpora, as linguistic resources, are indispensable in any NLP task. More and more researchers have come to realise how the quality of corpora would influence the performance of machine learning. The performance could be greatly improved if clean and well-formed corpora with consistent annotation are applied. Also, for different NLP tasks, we might require different kinds and types of corpora.

For Japanese morphological analysis, so far various Japanese corpora have been constructed and used in practice [3]. We found it hard to search for reliable information about all of the corpora and

[8] http://nlp.ist.i.kyoto-u.ac.jp/index.php?JUMAN++ retrieval date: 28/03/2018

	JUMAN	**ChaSen**	**MeCab**	**KyTea**	**JUMAN++**
Year of release	1992	1996	2002	2009	2016
Analysis model	bi-gram	variable-length-gram	bi-gram	variable-length-gram	RNNLM
Cost determination	manual	corpus	corpus	N/A	N/A
Learning model	Rules	HMM	CRF	SVM	RNNLM
Corpora trained on	RWCP Text Corpus	RWCP Text Corpus	Kyoto University Corpus & RWCP Text Corpus	Balanced Corpus of Contemporary Written Japanese	Kyoto University Text Corpus & Kyoto University Web Document Leads Corpus
N-best	No	No	Yes	Yes	No
Word segmentation	Yes	Yes	Yes	Yes	Yes
POS tagging	Yes	Yes	Yes	Yes	Yes
Lemmatisation	Yes	Yes	Yes	No	Yes
User training	Yes?	No?	Yes	Yes	Yes

Table 4: Comparison of traditional and RNN-based Japanese morphological analysers: JUMAN, ChaSen, MeCab, Kytea and JUMAN++

dictionaries of our interest. Here, we tried our best to collect and list some existing corpora and dictionaries that can be useful for Japanese morphological analysis and some related downstream NLP tasks. Information that we are uncertain about is either marked ? or left blank.

Corpora (see Table 7 for detailed comparison):

- RWCP Text Corpus[9]
- Kyoto University Text Corpus[10]
- Kyoto University Web Document Leads Corpus[11]
- Universal Dependencies[12]
- ...

Dictionaries (see Table 8 for detailed comparison):

- UniDic[13]

[9] https:www.jaist.ac.jpprojectNLP_PortaldocLRlr-cat-e.html#jp:rwc RWCP Text Corpus retrieval date: 29/03/2018

[10] http:nlp.ist.i.kyoto-u.ac.jpENindex.php?Kyoto%20University%20Text%20Corpus retrieval date: 29/03/2018

[11] http://nlp.ist.i.kyoto-u.ac.jp/EN/index.php?KWDLC retrieval date: 29/03/2018

[12] http://universaldependencies.org retrieval date: 29/03/2018

[13] http://unidic.ninjal.ac.jp retrieval date: 29/03/2018

- IPAdic[14]
- NEologd[15]
- also some bilingual dictionaries that can be potentially useful for morphological analysis as well...

5 Experiment: performance of existing morphological analysers on Japanese lemmatisation

Why do evaluation:

Although various Japanese morphological analysers have been built, as they are not always trained on the same corpora or using the same dictionaries, it could be hard to compare their performance in a fair way. We believe a fair comparison between different analysers can help us better understand the pros and cons of different algorithms and architectures. Although a fairer comparison would be to train the analysers on the same corpora and to test on the same test set as well, due to the limitation of time, we will just use the same test set to evaluate the analysers we mentioned above.

Why evaluate on lemmatisation only:

There are some reasons why we want to especially focus on the performance on the lemmatisation task. We found that lemmatisation has hardly ever been used as an evaluation metric by existing papers as far as we concerned (except in [7]). We have seen papers describing the above Japanese morphological analysers reporting the performance of segmentation and POS tagging more often such as in [14] [27] [33]. The reason could be, the performance of lemmatisation might depend on that of segmentation and POS tagging. As segmentation and POS tagging are the two pre-requisites for lemmatisation, we believe that segmentation can guarantee the correct units to be lemmatised, and accurate POS tagging might help with disambiguation of the same surface form. That is why we think lemmatisation is a comparatively *higher-level* task which could reflect the performance on segmentation and POS tagging as well.

Test set for evaluation:

The test set we use is from the Universal Dependencies (UD) project[16]. There are three Japanese corpora in UD[17]: KTC, GSD and PUD, all built on treebanks. A conclusive comparison[18] of them is available. None of them has ever been used by any previous study. By using them, hopefully we can see how well the existing morphological analysers adapt to and performance on them, and how suitable the corpora themselves could be used as training or test data for developing morphological analysers in the future. The specific test set we use comes from GSD, which mainly contains data collected from blogs and news. Some incomplete statistics of the test set are shown in Table 5.

Morphological analysers to be evaluated:

The morphological analysers we intend to compare include: JUMAN, ChaSen, MeCab, and JUMAN++. We exclude KyTea as it does not support lemmatisation. We use the latest version of the

[14] https://osdn.net/projects/ipadic/ retrieval date: 29/03/2018

[15] https://github.com/neologd retrieval date: 30/03/2018

[16] http://universaldependencies.org/#universal-dependencies retrieval date: 31/03/2018

[17] http://universaldependencies.org/#japanese-treebanks retrieval date: 31/03/2018

[18] http://universaldependencies.org/treebanks/ja-comparison.html retrieval date: 31/03/2018

# of sentences	557
# of tokens	12,615

Table 5: Statistics of the test set from the Universal Dependencies Japanese GSD corpus for evaluating the existing morphological analysers (JUMAN, ChaSen, MeCab and JUMAN++) on lemmatisation: # of sentences: amount of sentences; # of tokens: amount of tokens

abovementioned analysers except for JUMAN++. JUMAN++ version 1.03 is used for comparison instead of version 2.0 because version 2.0 is still in the stage of pre-release. We use the Python wrapper for the above analysers to do lemmatisation.

Performance evaluation metric:

We report the results in terms of precision, recall and F1 score. The reason of choosing these three is because, the length of the golden standard (expected) lemma sequence might be different from that of the predicted output by the analysers. In the confusion matrix of True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN), the counts of True is the length of the expected output; the counts of Positive is the length of the predicted output; the counts of TP is the correct output. We concern about the counts of TP with which we can compute the precision, recall and F1 score using Equation (2) (3) and (4) respectively.

$$precision = \frac{TP}{P} \tag{2}$$

$$recall = \frac{TP}{T} \tag{3}$$

$$F1 = \frac{2 \times precision \times recall}{precision + recall} \tag{4}$$

Details of implementation:

For more details of implementation, the original code, the test set, the analysers, and the results can be found in this GitHub repository[19]. Some interesting and pending issues are involved in the comments and docstrings.

Results:

We failed to install the Python wrapper for ChaSen, so we were unable to achieve and evaluate its performance. The precision, recall, and F1 scores of MeCab, JUMAN and JUMAN++ on lemmatisation using the same test set are reported in Table 6 below. From Table 6, MeCab outperformed JUMAN and JUMAN++ by 0.07 in F1 score. The F1 scores for JUMAN and JUMAN++ are rather close to each other, with JUMAN++ outperforming JUMAN by 0.006.

Based on our minor error analysis in limited time, we conjecture that the low F1 scores of JUMAN and JUMAN++ might be largely due to the ambiguous ways of writing the lemmas of *noun/adjective-verbs* (形容動詞 which have POS of adjective verbs and nouns at the same time). For example, two ways of representing the lemma of 最高 are acceptable, i.e. 最高 and 最高だ. MeCab tended to output the former kinds of lemmas, which matched the golden standard lemmas in the UD test set; while

[19] https://github.com/shijieyao/INFR11113TNLP retrieval date: 01/04/2018

both JUMAN and JUMAN++ were prone to the latter way which has an extra だ following. Some other inconsistent ways of representing lemmas of some other POSs are considered as possible reasons as well.

Regarding running time, MeCab and JUMAN took very short time while JUMAN++ took way longer time - almost 150 times longer. The slow speed issue of JUMAN++ version 1.* was also reported by [25], but in the pre-released JUMAN++ version 2.*, the problem is said to be solved[20].

We realise that the comparison of the three analysers is still far from complete and fair. Due to limited time, these are the only results and analysis we are able to demonstrate. However, still there are some future directions that are potentially interesting to explore. Given enough time, we hope we can first do error analysis with more details regarding more aspects. Second, as the research of the RNNLM-based morphological analyser is still ongoing, we want to explore more possibilities of applying RNNLM to Japanese lemmatisation. We can extend its architectures in various aspects, such as using Long Short-term Memory (LSTM) [10] and GRU [6] which might help solve the vanishing gradient problem of RNN, or attention mechanism [5], etc. Or we can vary the training data or the way of feeding it into the network, which is more empirical. Another problem of our great interest is RNNLM's ability in disambiguating the same surface form. In Japanese, as the writing system is complicated, the same surface form in *hiragana* is able to represent multiple words written in *kanji* with different meanings. For example, the surface form かう can represent 買う and 飼う which have totally different meanings: buying and raising (pets). The past tense inflection of かう which is かった can have even more different lemmas apart from the two just mentioned above. This specific linguistic phenomenon is posing a challenge to current Japanese morphological analysers. Most of them can do nothing more than simply deconjugating the surface forms. Can RNNLM learn to disambiguate the different lemmas hidden under the same surface form? We hope we could answer the question based on empirical results from experiments soon.

	Precision	**Recall**	**F1 score**
JUMAN	0.826	0.793	0.809
MeCab	0.877	0.868	0.872
JUMAN++	0.833	0.798	0.815

Table 6: Precision, recall and F1 score of JUMAN, MeCab, and JUMAN++ on lemmatisation using the test set in Universal Dependencies GSD corpus

6 Conclusion

Morphological analysis is still an unsolved NLP problem especially regarding specific languages. Different languages, with their specific linguistic features, challenge the task to different degrees and from different aspects. In Japanese morphological analysis, the problem of ambiguous word boundaries, inconsistent annotation criteria, ambiguous surface forms of different lemmas, etc. are still pending issues. Various methods, from rules, Hidden Markov Model, Conditional Random Fields, to Support Vector Machine, have been applied to the task, and they more or less show some satisfactory results. Currently, neural-based methods, such as recurrent neural network language modelling, are showing their potential in dealing with Japanese morphological analysis. Although in the lemmatisation task

[20] https://www.slideshare.net/eiennohito/juman-v2-a-practical-and-modern-morphological-analyzer retrieval date: 01/04/2018

we evaluated, neural-based methods do not seem to outperform the analysers built on traditional methods, they are still very flexible to be modified or changed in order to be adaptive to Japanese lemmatisation. In the future, we hope to dig into the architecture of the neural network or vary the training data to observe any changes in the networks' performance. Hopefully by those experiments, we can open the blackbox of recurrent neural networks a little bit more.

References

[1] Yukiko Sasaki Alam. A two-level morphological analysis of japanese. In *Texas Linguistic Forum*, volume 22, page 14Although, 1983.

[2] Rie Ando and Lillian Lee. Unsupervised statistical segmentation of japanese kanji strings. Technical report, Cornell University, 1999.

[3] Masayuki Asahara. Corpus-based japanese morphological analysis. *Nara Institute of Science and Technology, Doctor's Thesis*, 2003.

[4] Zhenisbek Assylbekov, Rustem Takhanov, Bagdat Myrzakhmetov, and Jonathan N Washington. Syllable-aware neural language models: A failure to beat character-aware ones. *arXiv preprint arXiv:1707.06480*, 2017.

[5] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[6] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[7] Yasuharu Den, Junpei Nakamura, Toshinobu Ogiso, and Hideki Ogura. A proper approach to japanese morphological analysis: Dictionary, model, and evaluation. In *LREC*, 2008.

[8] Takeshi Fuchi and Shinichiro Takagi. Japanese morphological analyzer using word co-occurrence: Jtag. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 409–413. Association for Computational Linguistics, 1998.

[9] Masatsugu Hangyo, Daisuke Kawahara, and Sadao Kurohashi. Building a diverse document leads corpus annotated with semantic relations. In *Proceedings of the 26th Pacific Asia Conference on Language, Information, and Computation*, pages 535–544, 2012.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Daisuke Kawahara and Sadao Kurohashi. Case frame compilation from the web using high-performance computing. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 1344–1347, 2006.

[12] Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. Construction of a japanese relevance-tagged corpus. In *LREC*, 2002.

[13] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. Character-aware neural language models. In *AAAI*, pages 2741–2749, 2016.

[14] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, 2004.

[15] Sadao Kurohashi and Makoto Nagao. Building a japanese parsed corpus while improving the parsing system. In *Proceedings of The 1st International Conference on Language Resources & Evaluation*, pages 719–724, 1998.

[16] Sadao Kurohashi and Makoto Nagao. Building a japanese parsed corpus. In *Treebanks*, pages 249–260. Springer, 2003.

[17] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[18] Thang Luong, Richard Socher, and Christopher Manning. Better word representations with recursive neural networks for morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 104–113, 2013.

[19] Yuji Matsumoto. Japanese morphological analysis system chasen manual. *NAIST Technical Report, NAIST-IS-TR97007*, 1997.

[20] Yuji Matsumoto, Akira Kitauchi, Tatsuo Yamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka, and Masayuki Asahara. Morphological analysis system chasen version 2.2. 1 manual. *Nara Institute of Science and Technology*, 2000.

[21] Andrew McCallum and Wei Li. Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics, 2003.

[22] Tomáš Mikolov. Statistical language models based on neural networks. *Presentation at Google, Mountain View, 2nd April*, 2012.

[23] Tomáš Mikolov, Anoop Deoras, Daniel Povey, Lukáš Burget, and Jan Černockỳ. Strategies for training large scale neural network language models. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 196–201. IEEE, 2011.

[24] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[25] Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2297, 2015.

[26] Tetsuji Nakagawa, Taku Kudo, and Yuji Matsumoto. Morphological analysis using support vector machines and proposal of revision learning. *Journal of Information Processing*, 44(5):1354–1367, 2003.

[27] Graham Neubig, Yosuke Nakata, and Shinsuke Mori. Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 529–533. Association for Computational Linguistics, 2011.

[28] Fuchun Peng, Fangfang Feng, and Andrew McCallum. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*, page 562. Association for Computational Linguistics, 2004.

[29] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[30] Jun Zhou Shaosheng Cao, Wei Lu and Xiaolong Li. cw2vec: Learning chinese word embeddings with stroke n-gram information. 2018.

[31] Yanchuan Sim. A morphological analyzer for japanese nouns, verbs and adjectives. *arXiv preprint arXiv:1410.0291*, 2014.

[32] Charles Sutton and Andrew McCallum. *An introduction to conditional random fields for relational learning*, volume 2. Introduction to statistical relational learning. MIT Press, 2006.

[33] Koichi Takeuchi and Yuji Matsumoto. Hmm parameter learning for japanese morphological analyzer. In *Proceedings of the 10th Pacific Asia Conference on Language, Information and Computation*, pages 163–172, 1995.

[34] Clara Vania and Adam Lopez. From characters to words to in between: Do we capture morphology? *arXiv preprint arXiv:1704.08352*, 2017.

[35] Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657, 2015.

Appendix

	RWCP Text Corpus	**Kyoto University Text Corpus**	**Kyoto University Web Document Leads Corpus**	**Universal Dependencies**
Release	Suspended now	Restricted	Public	Public
Fee		126,000 JPY (840 GBP)	Free	Free
Domain	Government, politics, formal report in NLP, news articles	News articles	Web documents	News articles, Wiki, blogs
Annotation	word segmentation, POS, text category, dictionary, syntax		Morphology, named entities, dependencies, predicate-argument structures	word segmentation, POS, degree and polarity
Size (tokens)	train 802,954 test 96,393			total 402832
Systems trained on	JUMAN & ChaSen & MeCab	JUMAN++	JUMAN++	NONE

Table 7: Mainstream corpora for Japanese morphological analysis

	UniDic	**IPAdic**	**NEologd**	**JUMANdic**
Release	Yes	Yes	Yes	Yes
Fee	Free	Free	Free	Free
Size				
Corpus based on	BCCWJ	IPA corpus	Web	Kyoto Corpus
Systems used	MeCab & ChaSen	MeCab		MeCab

Table 8: Mainstream dictionaries for Japanese morphologial analysis