A Probabilistic Algorithm for Segmenting Non-Kanji Japanese Strings

Virginia Teller
Hunter College and the Graduate School
The City University of New York
695 Park Avenue
New York, NY 10021
vmthc@cunyvm.cuny.edu

Eleanor Olds Batchelder
The Graduate School
The City University of New York
33 West 42nd Street
New York, NY 10036
eobgc@cunyvm.cuny.edu

Abstract

We present an algorithm for segmenting unrestricted Japanese text that is able to detect up to 98% of the words in a corpus. The segmentation technique, which is simple and extremely fast, does not depend on a lexicon or any formal notion of what a word is in Japanese, and the training procedure does not require annotated text of any kind. Relying almost exclusively on character type information and a table of hiragana bigram frequencies, the algorithm makes a decision as to whether to create word boundaries or not. This method divides strings of Japanese characters into units that are computationally tractable and that can be justified on lexical and syntactic grounds as well.

Introduction

A debate is being waged in the field of machine translation about the degree to which rationalist and empiricist approaches to linguistic knowledge should be used in MT systems. While most participants in the debate seem to agree that both methods are useful, albeit for different tasks, few have compared the limits of knowledge-based and statistics-based techniques in the various stages of translation.

The perspective adopted in this paper represents one end of the rationalist-empiricist spectrum. We report a study that assumes almost no rule-based knowledge and attempts to discover the maximum results that can be achieved with primarily statistical information about the problem domain. For the task of segmenting non-kanji strings in unrestricted Japanese text, we found that the success rate of this minimalist method approaches 95%.

Background

In recent years, researchers in the field of natural language processing have become interested in analyzing increasingly large bodies of text. Whereas a decade ago a corpus of one million words was considered large, corpora consisting of tens of millions of words are common today, and several are close to 100 million words. Since exhaustively parsing such enormous amounts of text is impractical, lexical analyzers called part-of-speech taggers have been used to obtain information about the lexical, syntactic, and some semantic properties of large corpora. Automatic text tagging is an important first step in discovering the linguistic structure of large text corpora.

Probabilistic approaches to tagging have developed in response to the failure of traditional rule-based systems to handle large-scale applications involving unrestricted text processing. Characterized by the brittleness of handcrafted rules and domain knowledge and the intractable amount of work required to build them and to port them to new domains and applications, the rule-based paradigm that has dominated NLP and artificial intelligence in general has come under close scrutiny. Stochastic taggers are one example of a class of alternative approaches, often referred to as "corpus-based" or "example-based" techniques, that use statistics rather than rules as the basis for NLP systems.

A major decision in the design of a tagger is to determine exactly what will count as a word, and whether two sequences of characters are instances of the same word or different words (Brown et al. 1992). This may sound trivial — after all, words are delimited by spaces — but it is a problem that has plagued linguists for decades. For example, is *shouldn't* one word or two? Is *shouldn't* different from *should not*? If hyphenated forms like *rule-based* and *Baum-Welch* (as in *Baum-Welch algorithm*) are to count as two words, then what about *vis-à-vis*? The effects of capitalization must also be considered, as the following example shows:

> Bill, please send the bill. Bill me today or bill me tomorrow. May I pay in May?

In the first sentence *Bill* is a proper noun and *bill* is a common noun. In the second sentence *Bill* and *bill* are the same — both are verbs — while in the third sentence *May* and *May* are different — one is a modal and the other a proper noun.

These problems are compounded in Japanese because, unlike English, sentences are written as continuous strings of characters without spaces between words. As a result, decisions about word boundaries are

all the more difficult, and lexical analysis plays an important preprocessing role in all Japanese natural language systems. Before text can be parsed, a lexical analyzer must segment the stream of input characters comprising each sentence.

Japanese lexical analyzers typically segment sentences in two major steps, first dividing each sentence into major phrases called *bunsetsu* composed of a content word and accompanying function words, e.g. noun+particle, verb+endings, and then discovering the individual words within each phrase. Algorithms based on the longest match principle perform the bulk of the work (Kawada 1990). To extract a bunsetsu structure from an input string, the system first proposes the longest candidate that matches a dictionary entry and then checks whether the result agrees with the rules for bunsetsu composition. If the check fails, the system backtracks, proposes another shorter word, and checks the composition rules again. This process is repeated until the sentence is divided into the least number of bunsetsu consistent with its structure (Ishizaki et al. 1989).

Maruyama et al. (1988) describe a sentence analyzer that consists of five stages, each of which exploits a distinct kind of knowledge that is stored in the form of a set of rules or a table. The five stages are: segmentation by (1) character type, (2) character sequence, and (3) a longest matching algorithm; (4) a bottom-up parallel algorithm if stage 3 fails; and (5) compound-word composition. The first three lines in the transliterated example below illustrate what the procedure must accomplish:

input:	sisutemugabunobunkaisuru.
bunsetsu:	sisutemuga/buno/bunkaisuru.
words:	sisutemu-ga/bun-o/bunkai-suru.
meaning:	system-subj/sentence-obj/analyze-nonpast
translation:	A/The system analyzes a/the sentence.

Two recent projects at BBN have used rule-based lexical analyzers to construct probabilistic models of Japanese segmentation and part-of-speech assignment. Matsukawa, Miller, & Weischedel (1993) based their work on JUMAN, developed at Kyoto University, which has a 40,000 word lexicon and tags with a success rate of about 93%. They used hand-corrected output from JUMAN to train an example-based algorithm to correct both segmentation errors and part of speech errors in JUMAN's output. POST, a stochastic tagger, then selects among ambiguous alternative segmentation and part-of-speech assignments and predicts the part of speech of unknown words. Papageorgiou (1994) trained a bigram hidden Markov model to segment Japanese text using the output of MAJESTY, a rule-based morphological preprocessor (Kitani & Mitamura 1993) that is reported to segment and tag Japanese text with better than 98% accuracy. Papageorgiou's method uses neither a lexicon of Japanese words nor explicit rules, basing its decisions instead solely on whether a two-character sequence is deemed more likely to continue a word or contain a word boundary. This approach was able to segment 90% of the words in test sentences correctly, compared to 91.7% for the JUMAN-based method.

Characteristics of Japanese Text

Japanese text is composed of four different types of characters: kanji characters borrowed more than a millennium ago from Chinese; two kana syllabaries, hiragana and katakana; and romaji, consisting of Roman alphabetic and Arabic numeral characters. The syllabaries contain equivalent sets of around 80 characters each. Hiragana is used for Japanese words and inflections, while katakana is used for words borrowed from foreign languages and for other special purposes. Lunde (1993:4) describes the distribution of character types as follows:

> Given an average sampling of Japanese writing, one normally finds 30 percent kanji, 60 percent hiragana, and 10 percent katakana. Actual percentages depend on the nature of the text. For example, you may find a higher percentage of kanji in technical literature, and a higher percentage of katakana in the literature of fields such as computer science, which make extensive use of loan words written in katakana.

The variable proportions of character types can easily be seen in a comparison of three different samples of Japanese text. The first corpus consists of a set of short newspaper articles on business ventures from *Yomiuri*. The second corpus contains a series of editorial columns from *Asahi Shinbun* (*tenseijingo shasetsu*, 1985-1991). Information on a third corpus was drawn from the description provided by Yokoyama (1989) of an online dictionary, *Shin-Meikai Kokugo Jiten*. Table 1 gives the size of each corpus in thousands of characters and shows the percentage of text written in each of the four character types. Punctuation and special symbols have been excluded from the counts.

	bus.	ed.	dict.
size (K chars)	42	275	2,508
% hiragana	30.2	58.0	52.4
% kanji	47.5	34.6	37.9
% katakana	19.3	4.8	6.8
% num/rom	2.9	2.6	2.9

Table 1

Of particular note is the fact that the business corpus contains roughly half the amount of hiragana of the other two samples, both of which come close to

Lunde's norm, and three to four times as much katakana. Table 2 lists the ten most frequent hiragana in the three corpora expressed as a percentage of total hiragana.

business		editorial		dictionary	
no	13.1	no	8.3	no	7.8
to	6.7	i	6.3	ru	6.3
ru	6.6	to	4.5	i	4.8
wo	6.5	ru	4.5	ni	4.4
ni	5.6	ni	4.5	to	4.1
ha	5.5	ta	4.4	wo	3.9
de	5.5	ha	4.1	na	3.5
si	5.3	ga	4.0	si	3.4
ta	4.4	wo	3.7	su	3.2
ga	3.6	na	3.5	ta	3.1

Table 2

Again, the business corpus exhibits characteristics that differ significantly from the editorial and dictionary samples. Although *no* is the most frequent hiragana in all three texts, it occurs almost twice as often in the business sample. Since one function of the particle *no* is to combine nouns into noun phrases, this result suggests there is a large amount of such compounding in business writing. In contrast, hiragana *i* and *na*, which appear in adjective inflections, are not found in the business top ten list, even though both are among the top ten in the other two corpora, and *i* is in the top three.

Defining a Word

Exactly what constitutes a word in Japanese for segmentation purposes is a controversial issue. Without spaces that delimit lexical units, the decision may be left largely to the designer of a particular segmentation method. The bunsetsu *BENKYOUsiteimasita* 'was studying', written with two initial kanji characters (shown in upper case below) and seven hiragana, can be considered a single lexical unit or can be divided into as many as six elements:

BEN+KYOU - si - te - i - ma+si - ta

containing the sequence:

'study' - 'do' - particle - progressive - polite - past

or into some other intermediate grouping. Because of this flexibility, the word boundaries produced by a particular segmentation method may vary from fairly large lexical units to small ones closer to a morphological level of analysis, and several positions along this spectrum can easily be defended. The consistency with which a segmenter makes its decisions is more important than the position taken on word boundaries. Systematic errors in output can be accounted for later in processing no matter what size units are produced.

The Segmentation Algorithm

The strategy underlying the design of the present segmentation algorithm was to discover the maximum results that could be achieved with a minimum of computational (and human) effort. To this end, the algorithm incorporates a simple statistical technique for segmenting hiragana strings into words, a measure that is loosely based on the notion of mutual information (Brill et al. 1990, Magerman & Marcus 1990).

During the first stage of processing, a program scans an input file of Japanese text and identifies each character as one of five types:

[h] hiragana
[K] kanji
[k] katakana
[P] punctuation and symbols
[R] romaji (Roman letters and Arabic numbers)

For each hiragana character the algorithm computes a bigram frequency count based on the type of character that immediately precedes and follows the hiragana.

Each hiragana character is tallied twice — once as a pair with its preceding hiragana character or other character type and once as a pair with the following character. The output of this stage of processing is a 90 x 90 bigram frequency array. The rows and columns in the array include 83 hiragana characters and 4 other character types plus an end count, an error count, and a row or column total. The end count is tallied whenever the hiragana character is the last in a string (the pair h + {K,k,P,R}).

The segmentation algorithm then uses the bigram frequency array previously computed to divide hiragana sequences in Japanese text into individual words. For each hiragana character a decision is made as to whether this hiragana begins a new word or is a continuation of the preceding one. The algorithm works as follows:

A. Hiragana characters that follow katakana, punctuation, or romaji characters are assumed to begin a new word. These cases fall into the category of "no decision needed."

B. A word boundary is created between two hiragana characters if the combined probability of the left character ending a word and the right one beginning a word is greater than a probability of the two occurring together. If the end/begin likelihood is equal to or less than the co-occurrence likelihood, no cut is made.

The likelihood of ending a word is estimated from the proportion of all occurrences of the hiragana character that immediately precede any non-hiragana character, i.e. the hiragana ends a hiragana string:

[h1 + {K,k,P,R}] / h1-total.

The assumption is that no word contains hiragana followed by non-hiragana. There are kanji compound "words", however, that are typically written with the first part in hiragana to represent a too-difficult kanji. Such compounds will be divided incorrectly by this method, as will the hiragana honorific prefixes before words written in kanji, e.g. *o+KANE*, *go+SENMON*.

The likelihood of beginning a word is estimated from the proportion of all occurrences of the hiragana character immediately following a character that is not kanji or hiragana:

$$[\{k,P,R\} + h2] / h2\text{-total}.$$

This measure is not completely convincing, because it omits the most frequent case of hiragana words, namely, where particles follow kanji. However, since these cases cannot automatically be distinguished from other cases (also numerous) where kanji+hiragana represent a single morpheme, the K+h2 count is omitted from the measure.

The likelihood of co-occurrence is estimated from the product of two percentages:

$$([h1+h2] / h1\text{-total}) * ([h1+h2] / h2\text{-total}).$$

This measure is also flawed due to the existence of certain highly frequent combinations that are not usually considered to be a single word, e.g. *de ha*, and others that are infrequent but undoubtedly a single word, e.g. *mono*.

C. Deciding whether to create a word boundary between a kanji character and a following hiragana presents the greatest difficulty, because some kanji-hiragana transitions are continuations of the same word (*YO+bu*) while others are not (*HON+de*). Division is based on a comparison of the frequency of the hiragana following kanji and its frequency following other non-hiragana characters in the set {k,P,R}. Our reasoning is that the non-kanji cases cannot be continuations (case A above), while the kanji cases can be either continuous or discontinuous. Four situations arise:

1. If this hiragana very rarely appears following non-kanji characters (h < 0.5% of all hiragana in post-kPR position), then its occurrence following kanji is assumed to be a continuation of the same word.
2. If this hiragana appears after non-kanji characters significantly more often (> 0.5%) than after kanji characters, then begin a new word.
3. Conversely, if the post-kanji ratio is greater than or equal to the post-kPR ratio, and the post-kPR ratio is less than 1%, then consider the hiragana a continuation.
4. Otherwise, if the probability of this hiragana beginning a word is greater than the probability that it is a continuation, then separate.

Results

Experiment 1

We conducted an initial experiment (Teller & Batchelder 1993) to assess the accuracy of the segmentation algorithm using the business corpus, which is a collection of 216 short texts averaging 6 to 7 lines each and totaling 1457 lines and 49,024 characters. In the experiment, 90% of the corpus was used to build the bigram frequency table, and the segmentation algorithm was tested on the remaining 10%. This corpus produced a sparse matrix with a total of 20,012 pairs tallied in 802 of the 7744 cells. Table 3 shows a fragment of this array that clearly reveals three high frequency hiragana strings: *kara*, a particle; *kiru*, a verb form (as in *dekiru*); and *kore*, a pronoun.

	ra	ri	ru	re	ro
ka	140	3	2	1	
ki			43	1	
ku	1	3	2		
ke	7	12		3	
ko				38	12

Table 3

The algorithm performed well in some respects and poorly in others. Although kanji compound verbs (kanji followed by *suru*) were correctly maintained as a unit, the treatment of the *-teiru/-deiru* stative/progressive verb ending was inconsistent. The *-te* form was left intact (te i ru) while the *-de* version was incorrectly segmented as de | i ru. The particles *nado*, *mo*, and *he* were not separated from preceding kanji, but the words *tomo* 'together' and *mono* 'thing' were divided in the middle. Some common adverbial phrases were joined and some were not. For example, *ni tsuite* 'concerning' was treated as a single word ni tsu i te, but the phrase *sude ni* 'already' was broken into su | de | ni.

Table 4 gives examples of correct and incorrect segmentation and suggests an improved segmentation for the incorrectly divided strings. A blank between two characters indicates they are part of the same word, while a '|' indicates a word boundary, and upper case denotes kanji.

correct
KA ri ru | ko to | ga | de ki ru | to i u
ko re | ma de | no
incorrect
ni | to | do | ma t te i ta
mo | no | de | ha | ka | na ri
better
ni | to do ma t te i ta
mo no | de | ha | ka na ri

Table 4

An analysis of the output when the test corpus was run revealed that 90.7% of the 697 hiragana strings were divided correctly, and 9.3% were divided incorrectly. A breakdown of the results is shown in Table 5.

category	strings
no decision needed: {k,P,R} + h	110
segmented correctly	522
segmented incorrectly	59
questionable decisions	6
total	697

Table 5

Experiment 2

The segmentation procedure was run recently on samples of the much larger editorial corpus. The portion of this corpus that we used to construct the bigram frequency array comprises 1.17 million characters, including punctuation and headers, of which 597,500 characters or 51% are hiragana. The 211,303 hiragana strings in the training corpus vary in length from 1 to 32 with an average length of 2.8. The hiragana portion of the bigram table (88 x 88) contains 808,803 entries in 3,916 cells, indicating that 51% of all possible hiragana pairs were encountered during processing.

When the segmentation algorithm was applied to a test corpus, it became obvious that additional training had produced a tendency to overdivide; the algorithm now preferred divisions to combinations. In order to constrain this tendency, two rules were added to the procedure:

1. Since the hiragana character *wo* is a specialized character that functions only as the object marking particle, a word boundary should always be placed on either side of it.

2. Eleven hiragana characters, including the most common postpositional particles, can occur singly as a word. No other hiragana characters are treated in this way.

A third proposed rule was eliminated after it was found not to affect the results significantly. This rule stated that small (subscripted) hiragana are always in the middle of a word and should suppress word boundaries on either side.

In addition, case A of the algorithm described above was modified so that a word boundary would automatically be created whenever a character type transition was encountered unless the transition involved kanji+hiragana, which is handled by case C. This change enabled us to evaluate the algorithm's ability to segment strings of any character type, including kanji. Kanji, katakana, and romaji strings are still left intact; only hiragana strings can be further divided or combined with preceding kanji. Nonetheless, our assumption is that this is the most appropriate choice for the vast majority of such strings.

With these enhancements, the segmentation procedure was rerun on a corpus of 2,200 characters containing the following proportions of character types: hiragana, 54%; kanji, 38%; katakana, 5%; numbers, 3%. (There was no romaji in this sample.) The segmented corpus was divided into 1172 strings, 570 or 49% of which were resolved on the basis of character type transitions alone. The algorithm inserted 602 additional boundaries, resulting in a total of 1172 words.

Assessing the accuracy of these results raises the difficult question of what to count as an error, given that the definition of a word in Japanese remains indeterminate. Word boundaries that separate stems and roots from inflectional and derivational endings cannot legitimately be described as errors for the reasons explained earlier. Consequently true errors must be those cases in which segmentation violates morpheme boundaries. One group of morphemes in the test corpus were wrongly divided because they contained statistically unusual hiragana sequences that could only have been identified as a unit by consulting a lexicon. A second class of errors occurred when the algorithm either separated two indivisible morphemes or divided a combination of two morphemes in the wrong place. These two types of failures to respect morpheme boundaries are illustrated below. The incorrect segmentation appears first, followed by the preferred version and a description of the sequence:

ta to I e ba	ta to e ba	(adv.)
to I te mo	to te mo	(adv.)
tsu mo I ri ra I shi i	tsu mo ri I ra shi i	(n. + adj.)
ki bi I ki bi	ki bi ki bi	(adv.)
ka I ke tsu ke te	ka ke I tsu ke te	(v. + v.)
de I ki ru	de ki ru	(v.)

Using this scoring method, the 1172 strings found by the segmenter contained 1106 correct words and 66 errors for an overall accuracy of 94.4%. The corpus actually contained 1125 words, so the fact that 1106 of these words were correctly identified amounts to a recall of 98.3%, and the precision, measured as the proportion of identified words that were correct (1106 of 1172) is 94.4%.

Conclusion

The method we have proposed for segmenting non-kanji strings has several strengths. It does not depend on a lexicon or even on any formal notion of what constitutes a word in Japanese, and the training phase does not require manually or automatically annotated text of any kind. In addition the technique is simple and extremely fast. Relying solely on character type information and hiragana bigram frequencies, the

algorithm makes a decision as to whether to create a word boundary or not. Moreover, we found that adding a log function to the computation, which makes the measure equivalent to the mutual information statistic, did not significantly change the results. This suggests that the extra work involved in computing mutual information may not be needed for the problem of segmenting non-kanji strings.

The robust performance of the segmentation algorithm is not surprising, because research has shown (see Nagao 1984) that character type information alone can be used to segment Japanese into bunsetsu units with about 84% accuracy. Our method improves on this result significantly, but we have purposely avoided dealing with the problems associated with segmenting strings of kanji. Work by Fujisaki and others (Fujisaki et al. 1991, Nishino & Fujisaki 1988, Takeda & Fujisaki 1987), however, has demonstrated that *n*-gram modeling techniques can be successfully applied to these more difficult cases.

The method, of course, has limitations as well. Without a lexicon it is virtually impossible to identify words that are composed of infrequent sequences of hiragana, for example. This is a problem shared by most probabilistic approaches to natural language processing. Furthermore, the algorithm is sensitive to the corpus characteristics in that it will perform better on a corpus with shorter rather than longer kanji strings.

One purpose in reporting this study has been to make explicit some of the difficulties associated with processing Japanese text. It is a mistake to assume that an approach that works well for English will work equally well for Japanese without modification. This is evident when one tries to apply the notion of what a word is in English to Japanese. Various groups have tackled similar problems and have reported success in dealing with them without always making clear the criteria by which such success should be judged. By describing our procedures in detail and pointing out, with examples, areas of failure as well as areas of success, we hope to contribute to what should be an ongoing debate that addresses these issues.

Acknowledgements

This work was supported in part by NSF grants IRI-8902106 and CDA-9222720 and by PSC-CUNY awards 6-69283, 6-63295 and 6-64277. Hartvig Dahl, Ted Dunning, Bill Gale, Hitoshi Isahara, and Fumiko Ohno provided valuable assistance.

References

Brill, E.; Magerman, D.; Marcus, M.; and Santorini, B. 1990. Deducing linguistic structure from the statistics of large corpora. In Proceedings of the DARPA Speech and Natural Language Workshop.

Brown, P.; Della Pietra, A.; Della Pietra, V.; Lafferty, J; and Mercer, R. 1992. Analysis, statistical transfer, and synthesis in machine translation. In Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation, 83-100.

Fujisaki, T.; Jelinek, F.; Cocke, J.; Black, E.; and Nishino, T. 1991. A probabilistic parsing method for sentence disambiguation. In *Current Issues in Parsing Technology*, M. Tomita, ed., 139-152. Boston: Kluwer Academic.

Ishizaki, S.; Sakamoto, Y.; Ikeda, T.; and Isahara, H. 1989. Machine translation systems developed by the Electrotechnical Laboratory of Japan. In *Future Computing Systems*, Vol. 2, No. 3, 275-291. Oxford University Press and Maruzen Company Limited.

Kawada, T. 1990. Inputting Japanese from the keyboard. *Journal of Information Processing* 13:10-14.

Kindaichi, K.; Kindaichi, H.; Kenbou, H.; Shibata, T.; and Yamada, T. eds. 1981. *Shin-Meikai Kokigo Jiten (New Concise Japanese Dictionary)*. Tokyo: Sanseido.

Kitani, T., and Mitamura, T. 1993. Japanese preprocessor for syntactic and semantic parsing. In Proceedings of the Conference on Artificial Intelligence Applications, 86-92.

Lunde, K. 1993. *Understanding Japanese Information Processing*. Sebastopol, CA: O'Reilly.

Magerman, D., and Marcus, M. 1990. Parsing a natural language using mutual information statistics. In Proceedings of the Eighth National Conference on Artificial Intelligence, 984-989.

Maruyama, N.; Morohashi, M.; Umeda, S.; and Sumita, E. 1988. A Japanese sentence analyzer. *IBM Journal of Research and Development* 32:238-250.

Matsukawa, T.; Miller, S.; and Weischedel, R. 1993. Example-based correction of word segmentation and part of speech labelling. In Proceedings of the ARPA Human Language Technology Workshop.

Nagao, M. ed. 1984. *Japanese Information Processing*. Tokyo: Denshi Tsuushin Gakkai. (in Japanese)

Nishino, T., and Fujisaki, T. 1988. Probabilistic parsing of Kanji compound words. *Journal of the Information Processing Society of Japan* 29,11. (in Japanese)

Papageorgiou, C. 1994. Japanese word segmentation by hidden Markov model. In Proceedings of the ARPA Human Language Technology Workshop, 271-276.

Takeda, K., and Fujisaki, T. 1987. Segmentation of Kanji primitive words by a stochastic method. *Journal of the Information Processing Society of Japan* 28,9. (in Japanese)

Teller, V., and Batchelder, E. 1993. A probabilistic approach to Japanese lexical analysis. AAAI Spring Symposium on Building Lexicons for Machine Translation. AAAI Technical Report SS-93-02.

Yokoyama, S. 1989. Occurrence frequency Ddta of a Japanese dictionary. In *Japanese Quantitative Linguistics*, S. Mizutani ed., 50-76. Bochum: Brockmeyer.