Analyzer to Identify Phrases and the Functional Roles in Sentences: Its Architectural Aspects*

Yukiko Sasaki Alam

Department of Digital Media, Hosei University
3-7-2 Kajino-cho, Koganei, Tokyo 184-8584, Japan
sasaki@hosei.ac.jp

Abstract. This paper presents the architectural aspects of the phrase analyzer that attempts to recognize phrases and identify the functional roles in the sentences in formal Japanese documents. Since the object of interest is a phrase, the current system, designed in an object-oriented architecture, contains the *Phrase* class, and makes use of the linguistic generalization about languages with Case markers that a phrase, whether a noun phrase, a verb phrase, a postposition (or preposition) phrase or a clause phrase, can be separated into the content and the function components. Without a dictionary, and drawing on the orthographic information on the words to parse, it also contains a class that identifies the types of characters, a class representing grammar, and a class playing the role of a controller. The system has a simple and intuitive structure, externally and internally, and therefore is easy to modify and extend.

Keywords: Phrase analyzer, Morphological analyzer, Functional roles of phrases, Japanese

1. Introduction

This paper describes the architecture of the small-scale phrase analyzer that attempts to parse into phrases texts transcribed wherever applicable in *Joyo Kanji* (frequently used Chinese characters), and to identify the functional roles of the phrases in the sentences. Such texts are found in articles in newspapers, professional magazines and journals, and government documents. Because there are no spaces between words in Japanese texts, word-breaking is not a straight-forward task. In the past there have been several morphological analyzers: notably, *Juman* (Kurohashi and Nagao, 2003) and *Chasen* (Matsumoto et al., 2000). The two large-scale analyzers parse Japanese texts into morphemes such as prefixes, suffixes, inflections, Case markers, particles and the components of compound words. The main task of the proposed system, however, is to parse sentences into phrases (not morphemes), and identify the functional roles of the phrases, because the objective is to understand sentences in terms of the functional roles of the phrases.

The second difference of the current system is that it is not intended to parse into phrases per se, but to identify the functional roles of the phrases. To recognize phrases, it resorts to a general characteristic of phrases in sentences found in formal documents. The general feature is that the component representing the content of a phrase is transcribed in *Kanji* (Chinese characters) or/and *Katakana* (phonetic characters used for transcribing words of foreign origin), and that the component expressing the function in *Hiragana* (phonetic characters used for transcribing words or morphemes of Japanese origin). There have been developed such morphological analyzers that exploit the orthographic difference between the content and

functional components of phrases: to name a few, Asahara (2003), Kazama (2001), Kashioka et al (1998), and Kameda (1996). All of them, unlike the current system, focus on the parsing into morphemes or phrases, but not on the identification of the functional roles of the phrases.

The third feature different from other systems is that it is purely rule-based, unlike Kudo (2002), Sekine (2001), Uchimoto (2002), Kanayama et al (2000), and Haruno et al. (1999), all of which are statistically modeled.

To give an idea of an output parsed by the current system, an example is given in Table 1.

Table 1: A successful output.

05年度は中途採用を当初計画の20人から最大150人まで拡大。
zero-go-nen-do-wa-chuuto-saiyo-o-tousho-keikaku-no
(0-5-fiscal-year-Topic-midway-employment-Object-initial-plan-of)
-nijuu-nin-kara-saidai-hyakugojuu-nin-made-kakudai
(-20-person-from-at-most-150-person-to-expansion)

内容	機能	文法的役割
05年度	は	話題 (TOPIC)
中途採用	を	目的語 (OBJECT)
当初計画	の	名詞修飾句 (NOMINAL MODIFIER)
20人	から	始点 (POINT OF DEPARTURE)
最大150人	まで	継続終点 (UP-TO)
拡大	(省略)	名詞止め文 (NOUN-ENDING SENTENCE)

The first column in the table contains the content components, the second the functional components, the third the descriptions of the functional roles. The meaning of the parsed sentence is that in the fiscal year of 2005, (the company) expanded the number of midway employment from the initially planned 20 to 150.

The present paper first describes the components of the system and the relations among them, then discusses the pros and cons before ending with a short conclusion.

2. Architecture

The current system is structured in an object-oriented design, comprising four classes (or programs). The main is the *MorphAlgorithm* class, and it is supported by the remaining three classes: the *Phrase*, *Grammar* and *CharIdentifier* classes. The object-oriented design facilitates the creation of classes that represent concepts we are already familiar with, and thus helps us understand the functions of the classes and the interaction between them. Since the current system attempts to parse sentences into phrases, it has classes representing the parsing algorithm, the syntactic unit of phrase, and the grammar containing grammatical information. In addition, the system is provided with a class named *CharIdentifier* so that it is able to recognize the types of characters, because it parses texts transcribed in *Joyo Kanji* (frequently used Chinese characters) wherever applicable, and thus makes use of the orthographic difference in the transcription of the content and the functional parts of phrases. Figure 1 below shows a simplified state chart in UML (Unified Modeling Language) for illustrating the relations among the classes in the current system.

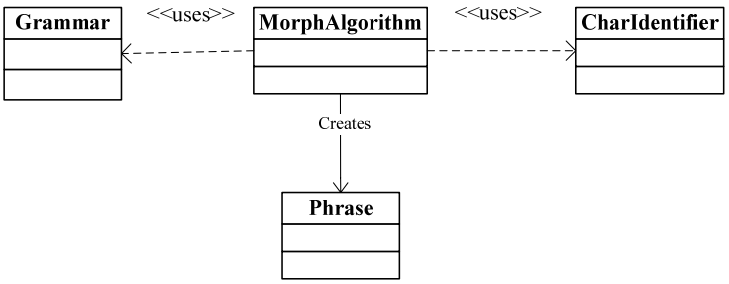

Figure 1: A simplified state chart diagram for the current system.

Below comes a more detailed description of each class in the current system.

2.1. The *MorphAlgorithm* class

The *MorphAlgorithm* class[1] is the main class of the present system that controls the process of recognizing phrases and identifying the functional roles played in the sentences of the text. The target texts are articles in newspapers, magazines, professional journals and public documents that are transcribed wherever applicable by using so called *Joyo Kanji* (frequently used Chinese characters). In such texts, the content parts of phrases are likely to be transcribed in *Kanji* (Chinese characters) or/and *Katakana* (phonetic characters used for transcribing words of foreign origin). At the same time most parts that indicate the functions of phrases are written in *Hiragana* (phonetic characters for transcribing words of Japanese origin). The system utilizes these orthographic features to recognize phrases, and identify the content and the functional parts. In addition, it resorts to another feature characteristic of Japanese (a head-final language) that the functional part is located at the end of a phrase.

The *MorphAlgorithm* class first cleans up the sentences to parse, for instance, by removing a tab stop and a new line characters. All the sentences in the text are appended into a *StringBuffer* instance, and are dealt with one sentence at a time. The algorithm that makes up the backbone of the *MorphAlgorithm* class (illustrated on Appendix A in this paper) is discussed in detail in Alam (2007). The following is a synopsis of the algorithm.

The algorithm first looks for a character or characters that are unacceptable at the beginning of a phrase. Such a character or characters are a period, a comma, a parenthesis and characters indicating a complementizer.[2] After examining these elements, the algorithm yet has to look for another irregular situation before going on to find a typical phrase with the content section consisting of *Kanji* or/and *Katakana*, followed by the functional *Hiragana* part. The irregular case is that a phrase, in particular, the content part begins with a *Hiragana* character. When it finds only one *Hiragana* character followed by a *Kanji* or a *Katakana*, it checks if the *Hiragana* is an honorific prefix. If not, and followed by another *Hiragana*, it keeps reading *Hiragana* characters until it hits a character that is not a *Hiragana*. Then the *Hiragana* sequence is assumed to be a phrase, and sent to the procedure in order to find the functional role of the phrase by its ending part.

When the algorithm does not find a *Hiragana* at the beginning of the phrase, but a *Kanji* or a *Katakana*, it keeps reading it until it meets a character that is not a *Kanji* or a *Katakana*. It

[1] The current system is written in Java programming language, in which a software system consists of many sets of programs called *classes*. Each *class* is a model that is a collection of procedures (called *methods*) and data (called *data fields*) used in the procedures. A *class* can be instantiated, and the instance can be used by another *class*. Some *classes* (often called *utility classes*) may never be constructed as instances, but the *methods* may be called and utilized by other *classes*.

[2] A *complementizer* is an element similar to *that* in *I think that many people like the movie*. と (pronounced 'toh') is a complementizer in Japanese. A complementizer cannot appear at the beginning of a phrase in Japanese, a Head-final language.

furthermore checks if the character that is not a *Kanji* or a *Katakana* is other than a *Hiragana* such as a period, a bracket or a comma. If it is, the sequence of *Kanji* or/and *Katakana* is a phrase that has the functional part omitted, and the phrase is marked accordingly (like an example presented in Table 1). When the sequence of *Kanji* or/and *Katakana* is followed by a *Hiragana* or a sequence of *Hiragana*, the phrase is probably a phrase typically found in a text targeted in the present system, and is sent to the procedure in order to find what functional role represented by the *Hiragana* or the sequence of *Hiragana*. As seen above, the *MorphAlgorithm* class is a class that controls the present system.

2.2. The *Phrase* class

The *Phrase* class represents a phrase, a syntactic unit that forms a sentence together with other phrases. A phrase is a group of words that are ofen used together and that have a special meaning and function. Among them are a noun phrase, a verb phrase, a preposition (or postposition) phrase, and a clause phrase. All the phrases can be separated into the content part that contains information on the meaning of the phrase, and the function part that indicates the functional role of the phrase played in the sentence. The function part is located at the end of a phrase in Japanese, a Head-final language. In formal Japanese documents, the content part tends to be transcribed in *Kanji* or/and *Katakana* while the function part in *Hiragana*. The current system has created the *Phrase* class by making use of these generalizations or likelihoods.

The *Phrase* class has many accessor methods, which are shown in Table 2. Whenever the algorithm in the *MorphAlgorithm* class detects the beginning boundary or the ending boundary of a phrase, that position on the sentence is recorded in the current instance of the *Phrase* class by calling the *setter* methods like *setHead* and *setNonhead*, listed in Table 2, and once the process of the current *Phrase* instance is complete, information contained in the instance is printed out by calling the *getter* methods such as *getHead* and *getNonhead*.

At present all the phrase instances processed in the sentence are not stored in a container of a *Collection* class, but once processed (or printed out) they are discarded. There might be a need in the future to store information in all the *Phrase* instances of a sentence, for instance, to identify an embedded clause such as a relative clause and a complementizer phrase or to disambiguate between adverbial and adjectival uses of preposition (or postposition) phrases. The current system can accomodate such a change, but that would require much of semantic study before implementation.

Table 2: Accessor methods in the *Phrase* class.

setters	getters
setHead	*getHead*
setNonhead	*getNonhead*
setGramRole	*getGramRole*
setHeadStart	*getNonheadStart*
setHeadEnd	*getHeadEnd*
setNonheadStart	*setNonheadStart*
setNonheadEnd	*getNonheadEnd*
setPhraseStart	*getPhraseStart*
setPhraseEnd	*getPhraseEnd*

During the process of looking for the beginning and the ending boundaries of the *Phrase* instance, the algorithm of the *MorphAlgorithm* class at the same time attempts to identify the grammatical role in terms of information provided by the *Grammar* class. Once obtained, information on the grammatical function of the *Phrase* instance is stored and printed out by calling the accessor methods in the *Phrase* instance.

2.3. The *Grammar* class

The *Grammar* class provides the *MorphAlgorithm* class with information on the functional roles of the function parts of phrases. For instance, Table 3 lists Case markers registered in the *Grammar* class.

Table 3: Case markers.

Case/ Particles	Pronun-ciation	Functional role(s)
が	ga	Subject marker
を	o	Object marker
は	wa	Topic marker
で	de	Place/Instrument/Conjunctive
へ	e	Goal
から	kara	Point of departure
まで	made	'up to/till'
より	yori	Point of departure (formal or archaic)

In addition, the *Grammar* class registers information on particles denoting conjunction (such as a particle denoting 'and'), clause particles that make up various clauses by immediately following sentences and propositions (such as a particle indicating 'because'), and particles denoting approximation (such as a particle indicating 'about'). See Alam (2007) for the list of all the grammatical information contained in the *Grammar* class.

Table 4 shows the names of the data fields (or data used in the methods) and the methods listed in the *Grammar* class. The prefix + indicates that the data field is a *public* data field while the prefix – a *private* one that is used only inside the *Grammar* class.

As the names of the data fields in Table 4 indicate, the functional particles, suffixes and inflections are first recognized by the number of *Hiragana* characters that make them up, and then are examined for the grammatical roles by calling the *getGrammaticalRole* method.

2.4. The *CharIdentifier* class

The *CharIdentifier* class is a utility class that helps identify the orthographic types of characters. The types identified by the class are *Kanji* (Chinese characters), *Katakana* (phonetic characters used for transcribing words of foreign origin), *Hiragana* (phonetic characters used for transcribing words of Japanese origin, Case markers, inflections, particles, etc.), Arabic numerals, the Roman alphabet, special symbols and punctuations. The methods used for identification are illustrated in Table 5.

Table 4: Data fields and methods in the *Grammar* class.

Grammar
+ONE_CHAR_GRAM_MARKERS +TWO_CHAR_GRAM_MARKERS +THREE_CHAR_GRAM_MARKER S +FOUR_CHAR_GRAM_MARKERS +FIVE_CHAR_GRAM_MARKERS -TWO_CHAR_PARTICLES -TWO_CHAR_VERBAL_SUFFIXES -THREE_CHAR_PARTICLES -THREE_CHAR_VERBAL_SUFFIXES -THREE_CHAR_ADJECTIVE_SUFFIXES -THREE_CHAR_CONJUNCTIVES -FOUR_CHAR_PARTICLES - FOUR_CHAR_VERBAL_SUFFIXES -FIVE_CHAR_PARTICLES -FIVE_CHAR_VERBAL_SUFFIXES -FIVE_CHAR_GRAM_PARTICLES -CONJ -ADV
+getGrammaticalRole() +isIn()

Table 5: Methods in the *CharIdentifier* class.

CharIdentifier
+isNumeric() +isKatakana() +isHiragana() +isKanji() +isValidChar() +isComma() +isPeriod() +isParen() +isOpenParen() +isCloseParen +isQuote() +isOpenQuote() +isCloseQuote() +isDateAlone() +isTimeSuffixAlone() +isTimeNoun()

3. Discussion

The current system, without a dictionary, is fairly successful in parsing texts transcribed wherever applicable by *Joyo Kanji* (frequently used Chinese characters), but it is not able to parse successfully sentences containing successive phrases transcribed in *Hiragana*, because it recognizes the beginning of a phrase by finding a character that is not a *Hiragana*. Therefore, it fails to parse correctly such texts as those found in books for young children that are written without using Chinese characters. For a better treatment, it must be enriched by finding a minimum heuristic way of handling phrases transcribed in *Hiragana* that appear successively. The current system also fails to recognize, for instance, nouns consisting of a mixture of *Kanji* and *Hiragana* characters. As a heuristic means, it would be a good idea to install a special list of such nouns so that they could be recognized as content words.

The present system is a simple and light-weight rule-based analyzer, thus parsing texts at a high speed. It could be embedded in a large-scale system, and be employed in a mode dealing with texts such as articles in newspapers and professional magazines. The larger-scale system, with a dictionary, can come into use whenever having phrases exclusively in *Hiragana*.

Verb phrases appear typically in the form of the verbal stem (or root) transcribed in *Kanji*, followed by the (suffixes and) tense inflection transcribed in *Hiragana*. At present, the current system is equipped with a heuristic means of handling verb phrases, for instance, by listing each possible form. It would be feasible, however, to parse verb phrases fairly accurately, and not in a heuristic manner, because the orders in which the verb stems and the verbal suffixes/inflections occur are predictable. Such a morphological analyzer of verb phrases is under preparation. Once completed, this engine could be incorporated into the current system without difficulty. A foreseeable problem would be to locate a verb phrase. In Japanese, which is a head-final language, verb phrases appear at the end of sentences. This information helps recognize a verb phrase found at the end of a sentence, but verb phrases can appear at the end

of clauses as well, for instance, at the end of relative clauses or complementizer phrases. For a better treatment, further linguistic study about clause boundaries would be needed.

Once a morphological analyzer of verb phrases is implemented in the present system, data fields in the *Grammar* class could get rid of much information on verbal suffixes and inflections, and eventually would contain a shorter list of functional elements to examine. As such data fields in the *Grammar* class are at present based on heuristics, the incorporation of a rule-based verb phrase morphological analyzer would transform the current system into a more principle-based system.

4. Conclusion

This paper has focused on the architecture of an analyzer to parse into phrases sentences in texts transcribed wherever applicable in *Joyo Kanji* (frequently used Chinese characters). Such texts are articles in newspapers, professional and technical magazines and government documents. The system is designed so that linguistic components are reflected in the programs. The design is simple and intuitive, and facilitates easy modification and extension. In spite of the limitations, this simple light-weight fast analyzer is fairly successful in recognizing phrases and identifying the functional roles played in the sentences. It could be used for many purposes: for instance, as a component in a larger system, as a tool preparing a dictionary, and as a preliminary analyzer of Japanese sentences.

References

Alam, Yukiko Sasaki. 2007. A Morpho-Syntactic Analyzer of Controlled Japanese. In T. H. King and E. M. Bender, ed., *Parallel Proceedings of GEAF 2007 Workshop*. Stanford, CA: CSLI Publications (CSLI Studies in Computational Linguistics ONLINE: http://csli-publications.stanford.edu/).

Asahara, Masayuki. 2003. *Corpus-based Japanese Morphological Analysis*. Ph.D. Thesis. Nara Institute of Science and Technology.

Fuchi, Takeshi and Shinichiro Takagi. 1998. Japanese Morphological Analyzer Using Word Co-occurrence. *Proceedings of the COLING*, pp. 409-413.

Haruno, Masahiko, Satoshi Shirai, and Yoshifumi Ooyama. 1999. Using Decision Trees to Construct a Practical Parser. *Machine Learning*, 34, 131-149.

Kameda, Masayuki. 1996. A Portable & Quick Japanese Parser: QJP. *Proceedings of the COLING*, pp. 616-621.

Kanayama, Hiroshi, Kentaro Torisawa, Yutaka Mitsuishi and Jun'ichi Tsujii. 2000. A Hybrid Japanese Parser with Hand-crafted Grammar and Statistics. *Proceedings of the COLING*, pp. 411-417.

Kashioka, Hideki, Yasuhiro Kawata and Yumiko Kinjo. 1998. Use of Mutual Information Based Character Clusters in Dictionary-less Morphological Analysis of Japanese. *Proceedings of the COLING*, pp. 658-662.

Kazama, Jun'ichi. 2001. *Adaptive Morphological Analysis with a Small Tagged Corpus*. Master Thesis. University of Tokyo.

Kurohashi, Sadao and Makoto Nagao. 2003. Building a Japanese Parsed corpus ― While Improving the Parsing System. In Anne Abeille, ed., *Treebank Building Using Parsed Corpora*, pp. 249-260. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Matsumoto, Yuji, Akira Kitauchi, TatsuoYamashita, Yoshitaka Hirano, Hiroshi Matsuda, Kazuma Takaoka and Masayuki Asahara. 2001. *Morphological Analysis System ChaSen version 2.2.4 Manual*. Nara, Japan: Nara Institute of Science and Technology.

Sekine, Satoshi. 2001. A Fast Japanese Sentence Analyzer. *Proceedings of the First International Workshop on MultiMedia Annotation*.

Suzuki, Hisami, Chris Brockett, and Gary Kacmarcik. 2000. Using a Broad-Coverage Parser for Word-Breaking in Japanese. *Proceedings of the COLING*, pp. 822-828.

Uchimoto, Kiyotaka, Masaki Murata, Satoshi Sekine and Hitoshi Isahara. 2000. Dependency Model Using Posterior Context. *Proceedings of the Sixth International Workshop on Parsing Technologies*, pp. 321-322.

Appendix: Algorithm employed in the *MorphAlgorithm* class (adopted from Alam 2007).