Using a Broad-Coverage Parser for Word-Breaking in Japanese

Hisami Suzuki, Chris Brockett and Gary Kacmarcik
Microsoft Research
One Microsoft Way
Redmond WA 98052 USA
{hisamis, chrisbkt, garykac}@microsoft.com

Abstract

We describe a method of word segmentation in Japanese in which a broad-coverage parser selects the best word sequence while producing a syntactic analysis. This technique is substantially different from traditional statistics- or heuristics-based models which attempt to select the best word sequence before handing it to the syntactic component. By breaking up the task of finding the best word sequence into the identification of words (in the word-breaking component) and the selection of the best sequence (a by-product of parsing), we have been able to simplify the task of each component and achieve high accuracy over a wide variety of data. Word-breaking accuracy of our system is currently around 97~98%.

1. Introduction

Word-breaking is an unavoidable and crucial first step toward sentence analysis in Japanese. In a sequential model of word-breaking and syntactic analysis without a feedback loop, the syntactic analyzer assumes that the results of word-breaking are correct, so for the parse to be successful, the input from the word-breaking component must include all words needed for a desired syntactic analysis. Previous approaches to Japanese word segmentation have relied on heuristics- or statistics-based models to find the single most likely sequence of words for a given string, which can then be passed to the syntactic component for further processing. The most common heuristics-based approach utilizes a connectivity matrix between parts-of-speech and word probabilities. The most likely analysis can be obtained by searching for the path with the minimum connective cost (Hisamitsu and Nitta 1990), often supplemented by additional heuristic devices such as the longest-string-match or the least-number-of-*bunsetsu* (phrase). Despite its popularity, the connective cost method has a major disadvantage in that hand-tuning is not only labor-intensive but also unsafe, since adjusting the cost for one string may cause another to break. Various heuristic (e.g. Kurohashi and Nagao 1998) and statistical (e.g. Takeuchi and Matsumoto 1997) augmentations of the minimum connective cost method have been proposed, bringing segmentation accuracy up to around 98-99% (e.g. Kurohashi and Nagao 1998, Fuchi and Takagi 1998).

Fully stochastic language models (e.g. Nagata 1994), on the other hand, do not allow such manual cost manipulation and precisely for that reason, improvements in segmentation accuracy are harder to achieve. Attaining a high accuracy using fully stochastic methods is particularly difficult for Japanese due to the prevalence of orthographic variants (a word can be spelled in many different ways by combining different character sets), which exacerbates the sparse data problem. As a result, the performance of stochastic models is usually not as good as the heuristics-based language models. The best accuracy reported for statistical methods to date is around 95% (e.g. Nagata 1994).

Our approach contrasts with the previous approaches in that the word-breaking component itself does not perform the selection of the best segmentation analysis at all. Instead, the word-breaker returns all possible words that span the given string in a word lattice, and the best word sequence is determined by applying the syntactic rules for building parse trees. In other words, there is no task of selecting the best segmentation per se; the best word-breaking analysis is merely a concomitant of the best syntactic parse. We demonstrate that a robust, broad-coverage parser can be implemented directly on a word lattice input and can be used to resolve word-breaking ambiguities effectively without adverse performance effects. A similar model of word-breaking is reported for the problem of Chinese word segmentation (Wu and Jiang 1998), but the amount of ambiguity that exists in the word lattice is much larger in Japanese, which requires a different treatment. In the following, we first describe the word-breaker and the parser in more detail (Section 2); we then report the results of segmentation accuracy (Section 3) and the results of related experiments assessing the effects of the segmentation ambiguities in the word lattice to parsing (Section 4). In Conclusion, we discuss implications for future research.

2. Using a broad-coverage parser for word-breaking

The word-breaking and syntactic components discussed in the current study are implemented within a broad-coverage, multi-purpose natural language understanding system being developed at Microsoft Research, whose ultimate goal is to achieve deep semantic understanding of natural language[1]. A detailed description of the system is found in Heidorn (in press). Though we focus on the word-breaking and syntactic components in this paper, the syntactic analysis is by no means the final goal of the system; rather, a parse tree is considered to be an approximate first step toward a more useful meaning representation. We also aim at being truly broad-coverage, i.e., returning useful analyses irrespective of the genre or the subject matter of the input text, be it a newspaper article or a piece of e-mail. For the proposed model of word-breaking to work well, the following properties of the parser are particularly important.

- The bottom-up chart parser creates syntactic analyses by building incrementally larger phrases from individual words and phrases (Jensen et al. 1993). The analyses that span the entire input string are the complete analyses, and the words used in that analysis constitutes the word-breaking analysis for the string. Incorrect words returned by the word-breaker are filtered out by the syntactic rules, and will not make it into the final complete parse.

- All the grammar rules, written in the formalism of Augmented Phrase Structure Grammar (Heidorn 1975), are binary, a feature crucial for dealing with free word-order and missing constituents (Jensen 1987). Not only has the rule formalism proven to be indispensable for parsing a wide range of English texts, it is all the more critical for parsing Japanese, as the free word-order and missing constituents are the norm for Japanese sentences.

- There is very little semantic dependency in the grammar rules, which is essential if the grammar is to be domain-independent. However, the grammar rules are elaborately conditioned on morphological and syntactic features, enabling much finer-grained parsing analyses than just relying on a small number of basic parts-of-speech (POS). This gives the grammar the power to disambiguate multiple word analyses in the input lattice.

Because we do not utilize semantic information, we perform no semantically motivated attachment of phrases during parsing. Instead, we parse them into a default analysis, which can then be expanded and disambiguated at later stages of processing using a large semantic knowledge base (Richardson 1997, Richardson et al. 1998). One of the goals of this paper is to show that the syntactic information alone can resolve the ambiguities in the word lattice sufficiently well to select the best breaking analysis in the absence of elaborate semantic information. Figure 1 (see Appendix) shows the default attachment of the relative clause to the closest NP. Though this structure may be semantically implausible, the word-breaking analysis is correct.

The word-breaking component of our system is described in detail in Kacmarcik et al. (2000). For the purpose of robust parsing, the component is expected to solve the following two problems:

- Lemmatization: Find possible words in the input text using a dictionary and its inflectional morphology, and return the dictionary entry forms (lemmas). Note that multiple lemmas are often possible for a given inflected form (e.g. surface form かって (*katte*) could be an inflected form of the verbs かう (*kau* "buy"), かつ (*katu* "win") or かる (*karu* "trim"), in which case all these forms must be returned. The dictionary the word-breaker uses has about 70,000 unique entries.

- Orthography normalization: Identify and normalize orthographic variants. This is a non-trivial task in Japanese, as words can be spelled using any combination of the four character

[1] Japanese is one of the seven languages under development in our lab, along with Chinese, English, French, German, Korean and Spanish.

types (*hiragana*, *katakana*, *kanji* and roman alphabets). Other types of spelling variations are also frequent in Japanese (see Tables in Kacmarcik et al. 2000 for examples). It is therefore very important that the word-breaker identifies and normalizes these spelling variants.

The present model of word-breaking and parsing offers some desirable properties:

- The division of labor between the word-breaking and parsing components simplifies the task of the word-breaker tremendously, allowing it to focus on the sufficiently complex task of lemmatization and orthography normalization.
- Maintenance and consistency benefit from the fact that the grammar rules are maintained in only one place, i.e., in the rules for the parsing component. There is no separate set of rules exclusive to the task of word-breaking analysis.
- Undesired breaking results can be attacked separately in the word-breaking or syntactic components, depending upon the nature of the problem. Recall errors, in which desired words are missing, must be corrected in the word-breaking component, while the precision errors, in which a wrong word is preferred over a correct one, are corrected in the syntactic component. Fixing a recall error in the word-breaking component may result in a new precision error, but this can then be readily corrected separately in syntax. The error-correction process is thus substantially simpler than the model that needs to correct both recall and precision errors within a single component.

3. Accuracy results

We now turn to reporting experimental results using the current version of our system.

3.1 Test data

The test data used for this study were derived from three sets of corpora, each consisting of 5,000 randomly selected sentences from three major genres of text: an electronic encyclopedia[2], the *Nikkei* newspaper[3], and a collection of pocketbooks. To evaluate word-breaking accuracy (Section 3.3), we used a smaller subset of 1,500 sentences from these three genres (500 from encyclopedia, 500 from newspaper, and 500 from pocketbooks corpora). No manual changes were made in the test corpora, and the developers of the system had no previous exposure to them. The average sentence length is 49.02 characters. Table 1 summarizes the test corpora.

corpus	average sentence length	number of sentences
A (encyclopedia)	48.07 chars	5,000
B (newspaper)	46.99 chars	5,000
C (pocketbooks)	52.00 chars	5,000
total	49.02 chars (average)	15,000

Table 1. Summary of test corpora

3.2 Evaluation measures

To measure word-breaking accuracy, we used the standard metric of *recall* (*R*) and *precision* (*P*):

$$R = \frac{Matched}{Correct_{total}} \qquad P = \frac{Matched}{Output_{total}}$$

in which *Matched* is a number of words correctly returned by the word-breaker, $Correct_{total}$ is the number of words in the target corpora, and $Output_{total}$ is the total number of words returned by the word-breaker. Target word-breaking analyses for the test corpora were created manually by two native speakers of Japanese holding a degree in linguistics. The target word-breaking and tagging was chosen to maximize the retrieval of words in our dictionary rather than following an independent guideline. Since the parser currently uses extremely crude heuristics to rank multiple parses, the first analysis was chosen as the best word-breaking analysis in this experiment whenever the parser returned multiple parse analyses.

3.3 Breaking and POS-labeled recall and precision

Table 2 illustrates breaking recall and precision. In the breaking-only analysis, we considered the analysis to be correct if the desired character string

[2] Microsoft Encarta 98 Encyclopedia.

[3] Nihon Keizai Shimbun (1993-1994). Made available by the Linguistic Data Consortium, University of Pennsylvania.

was returned. POS-labeled accuracy, on the other hand, requires that the word be returned with one of nine correct POS-labels[4]. Because we used our own test corpora and because what counts as a word varies among word-breakers, it is difficult to cross-compare accuracy results with those of other word-breakers. With that caveat, the results are roughly comparable with those reported by others for both recall and precision. Further development on the grammar can be expected to improve the figures in the future. Stylistic and topical differences among different genres of corpora affected the accuracy only minimally.

		A	B	C
word-breaking only	recall	98.4	98.5	98.0
	precision	97.9	98.1	97.4
POS-labeled	recall	98.0	97.8	97.0
	precision	97.5	97.4	96.4

Table 2. Recall and precision of 3 x 500 sentence corpora (in %)

4. Word-breaking ambiguity and parser performance

At this point, there are a number of interesting questions we can ask regarding the nature of ambiguity in the input word lattice and its effects on parsing and word-breaking. How is the word-breaking accuracy related to the parser coverage as a whole? What are the effects of the word-breaking ambiguity on the parser precision and performance? How much ambiguity does the input word lattice contain to begin with? In this section, we discuss each of these questions in turn.

4.1 Parser coverage and word-breaking accuracy

Currently, the parser coverage at the sentence level is about 70% on the 15,000-sentence blind corpora mentioned above. The parser coverage is measured automatically as the percentage of sentences that received a final sentential parse, which is a good indicator of the coverage in terms of desired parses in our system (Gamon et al. 1997). About 30% of the input strings did not receive a final sentential parse, due to incomplete, fragmental input or to word-breaker or parser errors. Even when the input did not reach the sentential parse, however, the maximally spanning structure built by the parser is often useful, as in Figure 2 (see Appendix). For this reason, word-breaking accuracy is much higher than parser coverage.

4.2 Ambiguity in the lattice

Figure 3 shows the precision of the word-breaker in terms of how frequently the words returned by the word-breaker are used in the final successful parse. For this experiment, we used a corpus of 4,153 sentences for which the correct POS and segmentation analyses have been annotated, and for which the parser is known to return a correct analysis. The word-breaker recall for this corpus is therefore 100%. We measured the precision figures against two kinds of input word lattice: (1) a lattice containing both breaking and POS ambiguities; (2) a lattice containing only POS ambiguity. The latter kind of ambiguity also exists in parsing languages with white space between words. The figure shows that when there is only POS ambiguity in the input lattice, there are about 1.5 times more records in the input word lattice than the number of words in the final parse (i.e., about 62% of the candidate words are in the final parse). When both word-breaking and POS ambiguities are considered, the input lattice size is about three times the number of words used in the final parse (or about 36% of the words in the input lattice end up being used in the final parse). This ratio remains stable across various sentence lengths.

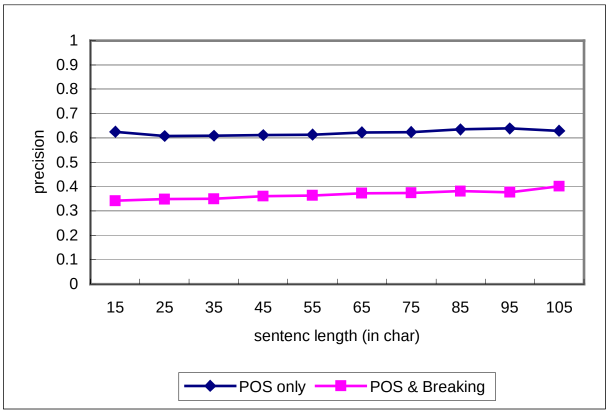

Figure 3. Precision of the word-breaker (y-axis) in relation to the sentence length (x-axis)

[4] 9 POSs include: Verb, Noun, Pronoun, Adjective, Adverb, Conjunction, Interjection, Postposition and special characters (e.g. punctuation marks and parentheses).

4.3 Parser precision

An initial concern in implementing the present model was that parsing ambiguous input might proliferate syntactic analyses. In theory, the number of analyses might grow exponentially as the input sentence length increased, making the reliable ranking of parse results unmanageable. In practice, however, pathological proliferation of syntactic analyses is not a problem[5]. Figure 4 tallies the average number of parses obtained in relation to sentence length for all successful parses in the 5,000-sentence test corpus (corpus A in Table 1). There were 4,121 successful parses in the corpus, corresponding to 82.42% coverage. From Figure 4, we can see that the number of parses does increase as the sentence grows longer, but the increment is linear and the slope is very moderate. Even in the highest-scoring range, the mean number of parses is only 2.17. Averaged over all sentence lengths, about 68% of the successfully parsed sentences receive only one parse, and 22% receive two parses. Only about 10% of sentences receive more than 2 analyses. From these results we conclude that the overgeneration of parse trees is not a practical concern within our approach.

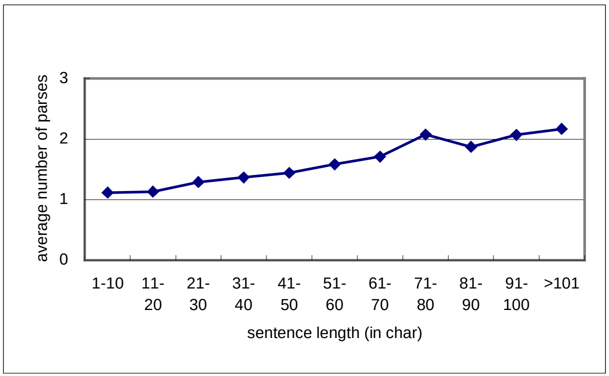

Figure 4. Average number of parses for corpus A (5,000 sentences)

4.4 Performance

A second potential concern was performance: would the increased number of records in the chart cause unacceptable degradation of system speed? This concern also proved unfounded in practice. In another experiment, we evaluated the processing speed of the system by measuring the time it takes per character in the input sentence (in milliseconds) relative to the sentence length. The results are given in Figure 5. This figure shows that the processing time per-character grows moderately as the sentence grows longer, due to the increased number of intermediate analyses created during the parsing. But the increase is linear, and we interpret these results as indicating that our approach is fully viable and realistic in terms of processing speed, and robust against input sentence length. The current average parsing time for our 15,000-sentence corpus (with average sentence length of 49.02 characters) is 23.09 sentences per second on a Dell 550MHz Pentium III machine with 512MB of RAM.

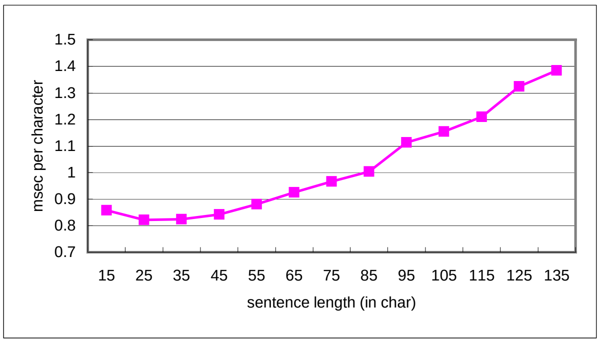

Figure 5. Processing speed on a 15,000-sentence corpus

5. Conclusion

We have shown that a practical, broad-coverage parser can be implemented without requiring the word-breaking component to return a single segmentation analysis, and that it can at the same time achieve high accuracy in POS-labeled word-breaking. Separating the tasks of word *identification* and best sequence *selection* offers flexibility in enhancing both recall and precision without sacrificing either at the cost of the other.

Our results show that morphological and syntactic information alone can resolve most word-breaking ambiguities. Nonetheless, some ambiguities require semantic and contextual information. For example, the following sentence allows two parses corresponding to two word-breaking analyses, of which the first is semantically preferred:

[5] A similar observation is made by Charniak et al. (forthcoming), who find that the number of final parses caused by additional POS tags is far less than the theoretical worst case in reality.

お茶にはいっているアルカロイドが効き目を発揮する。

(1) *ocha-**ni*** ***haitte**-iru arukaroido*
tea-in contain-ASP alkaloid
"the alkaloid contained in tea"

(2) *ocha-**ni-ha*** ***itte**-iru arukaroido*
tea-in-TOP go-ASP alkaloid
??"the alkaloid that has gone to the tea"

Likewise, the sentence below allows two different interpretations of the morpheme *de*, either as a locative marker (1) or as a copula (2). Both interpretations are syntactically and semantically valid; only contextual information can resolve the ambiguity.

来年はイスラエルである。

(1) *rainen-ha* *isuraeru-**de** aru*
next year-TOP Israel-LOC be-held
"It will be held in Israel next year".

(2) *rainen-ha* *isuraeru* ***de**-aru*
next year-TOP Israel be-PRES
"It will be Israel next year".

In both these sentences, we create syntactic trees for all syntactically valid interpretations, leaving the ambiguity intact. Such ambiguities can only be resolved with semantic and contextual information eventually made available by higher processing components. This will be the focus of our ongoing research.

Acknowledgements

We would like to thank Mari Brunson and Kazuko Robertshaw for annotating corpora for target word-breaking and POS tagging. We also thank the anonymous reviewers and the members of the MSR NLP group for their comments on the earlier version of the paper.

References

Charniak, Eugene, Glenn Carroll, John Adcock, Antony Cassandra, Yoshihiko Gotoh, Jeremy Katz, Michael Littman, and John McCann. Forthcoming. Taggers for parsers. To appear in *Artificial Intelligence*.

Fuchi, Takeshi and Shinichiro Takagi. 1998. Japanese Morphological Analyzer Using Word Co-Occurrence -JTAG-. Proceeding of ACL-COLING: 409-413.

Gamon, Michael, Carmen Lozano, Jessie Pinkham and Tom Reutter. 1997. Practical Experience with Grammar Sharing in Multilingual NLP. Jill Burstein and Claudia Leacock (eds.), *From Research to Commercial Applications: Making NLP Work in Practice* (Proceedings of a Workshop Sponsored by the Association for Computational Linguistics). pp.49-56.

Heidorn, George. 1975. Augmented Phrase Structure Grammars. In B.L. Webber and R.C. Schank (eds.), *Theoretical Issues in Natural Language Processing*. ACL 1975: 1-5.

Heidorn, George. In press. Intelligent Writing Assistance. To appear in Robert Dale, Hermann Moisl and Harold Somers (eds.), *Handbook of Natural Language Processing*. Chapter 8.

Hisamitsu, T. and Y. Nitta. 1990. Morphological Analysis by Minimum Connective-Cost Method. Technical Report, SIGNLC 90-8. IEICE pp.17-24 (in Japanese).

Jensen, Karen. 1987. Binary Rules and Non-Binary Trees: Breaking Down the Concept of Phrase Structure. In Alexis Manaster-Ramer (ed.), *Mathematics of Language*. Amsterdam: John Benjamins Publishing. pp.65-86.

Jensen, Karen, George E. Heidorn and Stephen D. Richardson (eds.). 1993. *Natural Language Processing: The PLNLP approach*. Kluwer: Boston.

Kacmarcik, Gary, Chris Brockett and Hisami Suzuki. 2000. Robust Segmentation of Japanese Text into a Lattice for Parsing. Proceedings of COLING 2000.

Kurohashi, Sadao and Makoto Nagao. 1998. Building a Japanese Parsed Corpus While Improving the Parsing System. First LREC Proceedings: 719-724.

Murakami, J. and S. Sagayama. 1992. Hidden Markov Model Applied to Morphological Analysis. IPSJ 3: 161-162 (in Japanese).

Nagata, Masaaki. 1994. A Stochastic Japanese Morphological Analyzer Using a Forward-DP Backward-*A** N-Best Search Algorithm. Proceedings of COLING '94: 201-207.

Richardson, Stephen D. 1997. Determining Similarity and Inferring Relations in a Lexical Knowledge Base. Ph.D. dissertation. The City University of New York.

Richardson, Stephen D., William B. Dolan and Lucy Vanderwende. 1998. MindNet: acquiring and structuring semantic information from text. Proceedings of COLING-ACL '98: 1098-1102.

Takeuchi, Koichi and Yuji Matsumoto. 1997. HMM Parameter Learning for Japanese Morphological Analyzer. IPSJ: 38-3 (in Japanese).

Wu, Andi and Zixin Jiang. 1998. Word Segmentation in Sentence Analysis. Technical Report, MSR-TR-99-10. Microsoft Research.

Appendix

あいつぐ隣国の干渉になやんでいた。

“(It) has been annoyed by the successive interventions by the neighboring country”.

```
-----------------------------------------------------
 あいつぐ VERB1
 あい NOUN1
 あい VERB2
:::::::: つぐ VERB3
:::::::::::::::: 隣国 NOUN2
:::::::::::::::::::::::: の NOUN3
:::::::::::::::::::::::: の PRON1
:::::::::::::::::::::::: の POSP1
:::::::::::::::::::::::::::: 干渉 NOUN4
:::::::::::::::::::::::::::::::::::: にな NOUN5
:::::::::::::::::::::::::::::::::::: に VERB4
:::::::::::::::::::::::::::::::::::: に POSP2
:::::::::::::::::::::::::::::::::::::::: なやんで VERB5
:::::::::::::::::::::::::::::::::::::::: なや NOUN6
:::::::::::::::::::::::::::::::::::::::: な VERB6
:::::::::::::::::::::::::::::::::::::::: な IJ1
:::::::::::::::::::::::::::::::::::::::: な POSP3
:::::::::::::::::::::::::::::::::::::::::::: やんで VERB7
:::::::::::::::::::::::::::::::::::::::::::: や IJ2
:::::::::::::::::::::::::::::::::::::::::::: や POSP4
:::::::::::::::::::::::::::::::::::::::::::::::: ん NOUN7
:::::::::::::::::::::::::::::::::::::::::::::::::::: で VERB8
:::::::::::::::::::::::::::::::::::::::::::::::::::: で CONJ1
:::::::::::::::::::::::::::::::::::::::::::::::::::: で POSP5
:::::::::::::::::::::::::::::::::::::::::::::::::::::::: いた NOUN8
:::::::::::::::::::::::::::::::::::::::::::::::::::::::: いた VERB9
-----------------------------------------------------
DECL1   NP1     NP2     RELCL1  VERB1*  "あいつぐ"   (successive)
                        NOUN2*   "隣国"              (neighboring country)
                        PP1     POSP1*   "の"        (GEN)
                NOUN4*  "干渉"                       (intervention)
                PP2     POSP2*   "に"                (by)
        VERB5*   "なやんで"                          (annoyed)
        AUXP1    VERB9*   "いた"                     (be)
        CHAR1    "。"
  ------------------------------
```

Figure 1. Example of ambiguous attachment. `RELCL1` can syntactically modify either `NOUN2` or `NOUN4`. `NOUN4` (non-local attachment) is the semantically correct choice. Shown above the parse tree is the input word lattice returned from the word-breaker.

タイ古典文学は伝統と歴史にもとづいている。”Classical Thai literature is based on tradition and history”.

```
FITTED1 NP1     NOUN1*  "タイ 古典 文学"    (classical Thai literature)
                PP1      POSP1*   "は"      (TOPIC)
        NP2     NP3      NOUN2*   "伝統"    (tradition)
                POSP2*   "と"               (and)
                NP4      NOUN3*   "歴史"    (history)
                PP2      POSP3*   "に"      (on)
        VP1     VERB1*   "もとづいて"       (based)
                AUXP1    VERB2*   "いる"    (be)
        CHAR1   "。"
  ------------------------------
```

Figure 2. Example of an incomplete parse with correct word-breaking results.