Lightweight Morphological Analysis Model for Smart Home Applications Based on Natural Language Interfaces

Sangwoo Kang, Harksoo Kim, Hyun-Kyu Kang, and Jungyun Seo

1 Department of Computer Science and Engineering, Sogang University, Seoul 121-742, Republic of Korea
2 Program of Computer and Communications Engineering, Kangwon National University, Chuncheon-si, Gangwon-do 200-701, Republic of Korea
3 Department of Computer Engineering, Konkuk University, Chungju-si, Chungcheongbuk-do 380-701, Republic of Korea

Correspondence should be addressed to Harksoo Kim; nlpdrkim@kangwon.ac.kr

Received 10 December 2013; Revised 19 February 2014; Accepted 8 March 2014; Published 23 June 2014

Academic Editor: Maria Jose Santofimia

Copyright © 2014 Sangwoo Kang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the rapid evolution of the smart home environment, the demand for natural language processing (NLP) applications on information appliances is increasing. However, it is not easy to embed NLP-based applications in information appliances because most information appliances have hardware constraints such as small memory, limited battery capacity, and restricted processing power. In this paper, we propose a lightweight morphological analysis model, which provides the first step module of NLP for many languages. To overcome hardware constraints, the proposed model modifies a well-known left-longest-match-preference (LLMP) model and simplifies a conventional hidden Markov model (HMM). In the experiments, the proposed model exhibited good performance (a response time of 0.0195 sec per sentence, a memory usage of 1.85 MB, a precision of 92%, and a recall rate of 90%) in terms of the various evaluation measures. On the basis of these experiments, we conclude that the proposed model is suitable for natural language interfaces of information appliances with many hardware limitations because it requires less memory and consumes less battery power.

1. Introduction

A smart home is a home in which all systems work together to make residents’ lives better with more control. In smart homes, household appliances are being rapidly evolved into information appliances (e.g., smartphones and personal digital assistants (PDAs)), which are usable for the purposes of computing, telecommunicating, reproducing, and presenting encoded information in myriad forms and applications. The information appliances will play important roles in the improvement of the quality of life, safety, and security as well as the communication possibilities with the outside world [1]. Therefore, future information appliances will interact with residents via social networking services (SNS) such as Twitter (http://www.twitter.com/), Facebook (http://www.facebook.com/), and Line (http://line.me/en/) [2, 3], as shown in Figure 1.

To implement such interactions via social networking, information appliances need to be enabled to a Web server or a gateway. Recent approaches have shown methods to embed Web servers directly in resource-constrained devices [2]. As shown in Figure 1, some information appliances in which embedded Web servers will be registered as users’ friends. Then, the registered information appliances will execute various commands that are received from users via social networking services. To realize such smart homes, information appliances should understand users’ natural language commands which are in the form of short text messages (e.g., tweets and recognized speech inputs) [2]. Natural language processing (NLP) techniques can be used to convert a natural language into a formal language that information appliances can understand [4], as shown in Figure 2.
the segmented words with part-of-speech (POS) tags. In
inflective languages, the major goal of the segmentation
process is to find roots of words (e.g., hours = hour+s/plural-
noun). In noninflective languages such as Chinese, the major
goal of the segmentation process is to correctly split a
compound word in a sequence of morphemes (e.g., 美國人 = 美國 (America) + 人 (people)). A named entity recognizer
groups some words into meaningful units (e.g., temperature
and time). A semantic and speech act analyzer generates a
machine-readable semantic form (e.g., set (temperature = 25,
time = 14:30)) and identifies user’s intention that is implied in
an input sentence (e.g., request). As shown in the NLP steps,
the initial step in the development of NLP-based applications
is to implement a high-performance morphological analyzer
(i.e., a morpheme segmentation and part-of-speech (POS)
tagging system). However, this implementation is not easy
because many information appliances have limited input and
output capabilities, limited bandwidth, limited memory, lim-
ited battery capacity, and restricted processing power. These
hardware limitations make it difficult to use the well-known
morphological analysis models that require complex compu-
tations on a large amount of training data. Although many
high-performance information appliances are developed at
present, lightweight morphological analyzers are still needed
to efficiently realize high-level NLP applications because
high-level linguistic models (e.g., named entity recognition,
semantic analysis, speech act analysis, and so on) require
large memory and high-performance processor. To resolve
this problem, we propose a morpheme segmentation and
POS tagging model that combines a rule-based method with
a statistical method. The current version of the proposed
system operates in Korean, but we believe that changing the
language will not be a difficult task because the system simply
uses a combination of widely used language-independent
NLP techniques such as a longest-matching method and a
hidden Markov model (HMM).

This paper is organized as follows. In Section 2, we review
the previous work on morpheme segmentation and POS
tagging systems. In Section 3, we present a hybrid system for morpheme segmentation and POS tagging in information appliances with restricted resources. In Section 4, we report the result of our experiments. Finally, we draw conclusions in Section 5.

2. Related Works

Morpheme segmentation and POS tagging have been widely studied by many researchers [5–8]. Previous morpheme segmentation methods can be classified into two groups: rule-based methods [9–12] and tabular parsing models [13]. Since the rule-based models are based on stemming [9, 10] or longest matching [11, 12], they are widely used for analytic languages (i.e., isolating languages) with low morpheme-per-word ratios (e.g., Chinese and English). Although rule-based models are simple and exhibit decent performance, they are not appropriate for synthetic languages (i.e., agglutinative languages) with high morpheme-per-word ratios (e.g., Korean, Japanese, and Turkish) because various linguistic problems occur in separating a word into a sequence of morphemes. Therefore, tabular parsing models are widely used for the Korean language, although they require complex computations to identify all possible morpheme candidates. However, it is impractical to use these tabular parsing models in information appliances, which typically have restricted processing power. To resolve this problem, we propose an efficient morpheme segmentation method based on modified longest-match-preference rules.

The initial approaches to POS tagging were based on rule-based models. Karlsson [14] applied constraint grammars (the grammar formalism was specified as a list of linguistic constraints) to POS tagging. Some researchers dealt with POS tagging as a part of syntactic analysis using rules that had been handcrafted on the basis of knowledge of morphology and intuition [15, 16]. Although these rule-based models are simple and clear, they have some drawbacks. First, they require handcrafted linguistic knowledge, which is considerably costly to construct and maintain. Second, they cannot effectively handle unknown word patterns because they use lexical levels of predefined patterns. Approaches that are designed to resolve these problems are mainly based on statistical models. The HMM is a representative model of statistical POS tagging for many languages [17]. To improve performance, some researchers have tried to apply effective smoothing methods or language-dependent characteristics to a conventional HMM [17, 18]. Because these statistical models can automatically obtain the necessary information for POS tagging, they do not require the construction and maintenance of linguistic knowledge. In addition, they are generally more robust to unknown word patterns than to the rule-based models. However, in information appliances with a small main memory, it is impractical to use these statistical models because they have large memory requirements. Conditional random fields (CRFs) and maximum entropy Markov models (MEMMs) are good frameworks that use contextual features for building probabilistic models to segment and label sequence data [19]. However, the strength of these discriminative models cannot help being restricted in information appliances with restricted processing power because they generally require more complex computations than an HMM for parameter estimations and probability calculations. Kudo et al. [20] proposed a compact CRF-based model in POS tagging of Japanese. Although Kudo’s model showed good performances, it still requires larger memory capacity than an HMM-based model because it uses additional n-gram features in order to increase performances. In the experiments on automatic word spacing which are performed in a commercial mobile phone with a XSCALE PXA270 CPU, 51.26 MB memory, and Windows Mobile 5.0, a CRF-based model was 2.11 times slower in response speed and 77.61 times larger in memory usage than an HMM-based model. To resolve these problems, we proposed a modified hidden Markov model that requires much less memory for loading statistical information.

3. Lightweight Morphological Analysis and POS Tagging

3.1. Modified Left-Longest-Match-Preference Method for Morpheme Segmentation

In English, a word is a spacing unit, but in Korean, an eojeol that consists of one or more morphemes comprises a spacing unit. Therefore, for morphological analysis of Korean sentences, eojeol’s should be first segmented into several morphemes. In this paper, we refer to an eojeol as a word for convenience because an eojeol plays the similar role as a word in the English language. To aid the readability of the examples, we use Romanized Korean characters called Hangul and insert hyphen symbols between Korean characters called eumjeol’s. The segmented morphemes can then be recovered into their lemma forms (i.e., lexical roots). To perform these processes in information appliances, we propose a method based on modified left-longest-match-preference (LLMP) rules. The conventional LLMP model scans an input word from left to right and matches the input word against each key in a morpheme dictionary. Then, it returns a lemma form of the longest-matched key and continues to scan the remainder of the input word. If a lemma has various POSs, the LLMP model assigns the most frequent POS to the lemma. Owing to the characteristic of longest matching, the conventional LLMP model cannot find all morpheme candidates in an input word, as shown in Figure 3.

In Figure 3, the correct morpheme sequence of “jip-gwon-han (doing the seizure of power)” is “jip-gwon (seizure of power)/noun + ha (do)/verb_suffix + n (-ing)/ending” in this context. However, the conventional LLMP model only returns “jip-gwon (seizure of power)/noun + han (hate)/noun” because “han (hate)/noun” is a longer morpheme than “ha (do)/verb_suffix” and “n (-ing)/ending”. We refer to short morphemes that are covered by long morphemes as hidden morphemes. To increase the recall rate of morphological analysis by resolving this hidden morpheme problem, we modify the LLMP model by adding supplementary rules for finding hidden morphemes. To construct the supplementary rules, we first implemented
Two morphemes generated by the conventional LIMP model

![Diagram showing morphemes](image)

**Figure 3: Example of the wrong left longest match.**

### Table 1: Subset of the decomposition rules.

| Target morpheme | Morpheme sequence | Frequency  |
|------------------|-------------------|------------|
| han              | ha (do)/adjective_suffix + n (-ing)/ending | 18206 (5.56%) |
| han              | ha (do)/verb_suffix + n (-ing)/ending | 9865 (3.01%) |
| ha-go            | ha (reason)/verb_suffix + go (and)/ending | 9494 (2.90%) |
| hal              | ha (do)/verb_suffix + l (-ing)/ending | 8816 (2.69%) |
| jeog-in          | jeog (having)/noun_suffix + i (be)/copula + n (-ng)/ending | 7615 (2.32%) |
| deul-i           | deul (-s)/noun_suffix + i (subject case)/propositional_word | 7085 (2.16%) |

A Korean morpheme segmentation system based on the LLMP model. Second, we annotated a large Korean corpus using the morpheme segmentation system. By comparing the results of automatic annotation with the correct results of human annotation, we automatically collected the cases where a long morpheme should be divided into a set of shorter morphemes. Finally, we selected the top-\(n\) cases that most frequently occurred and represented each case using symbolic rules, as listed in Table 1. We refer to these symbolic rules as decomposition rules.

By using the decomposition rules, the modified LLMP model adds hidden morphemes to the results obtained by the initial analysis, performed using a conventional LLMP model. For example, the modified LLMP model matches the longest-match morpheme “han (hate)” against “han (hate)” → “ha (do)/verb_suffix + n (-ing)/ending” and “han (hate)” → “ha (do)/adjective_suffix + n (-ing)/ending” in the decomposition rules. Then, it adds “ha (do)/verb_suffix + n (-ing)/ending” and “ha (do)/adjective_suffix + n (-ing)/ending” to the original morpheme sequence, as shown in Figure 4.

3.2. Simplified HMM for POS Tagging. Let \(W_{1,n}\) denote a sentence that consists of a sequence of \(n\) words, \(w_1, w_2, \ldots, w_n\), and let \(T_{1,n}\) denote the POS tag sequence, \(t_1, t_2, \ldots, t_n\), of \(C_{1,n}\).

The tagging problem can then be formally defined as finding \(T_{1,n}\), which results in

\[
T(W_{1,n}) \overset{\text{def}}{=} \arg \max_{T_{1,n}} P(T_{1,n} | W_{1,n}) = \arg \max_{T_{1,n}} \frac{P(T_{1,n}, W_{1,n})}{P(W_{1,n})} = \arg \max_{T_{1,n}} P(T_{1,n}, W_{1,n}).
\]

In (1), \(P(W_{1,n})\) is dropped as it is constant for all \(T_{1,n}\) terms. Next, (1) is broken into smaller pieces to collect statistics about each piece, as shown in

\[
P(T_{1,n}, W_{1,n}) = \prod_{i=1}^{n} P(w_i | t_{i-1}, w_{1:i-1}) P(t_i | t_{i-1}, w_{1:i-1}).
\]

Equation (2) is simplified by making two assumptions: the current POS tag is dependent only upon the previous POS tag and the current word is only affected by its POS tag. Equation (3) is a well-known HMM model for POS tagging:

\[
P(T_{1,n}, W_{1,n}) \approx \prod_{i=1}^{n} P(w_i | t_i) P(t_i | t_{i-1}).
\]
In (3), $P(w_i | t_i)$ and $P(t_i | t_{i-1})$ are called an observation probability and a transition probability, respectively. In Korean, it is difficult to calculate both the observation probability and the transition probability because a word generally consists of multiple morphemes. Therefore, many previous systems have in-word HMMs for calculating the observation probabilities of words, as shown in Figure 5.

In Figure 5, the gray rectangles represent the in-word HMMs based on the modified LLMP model. However, these in-word HMMs require more computing power, because they increase the complexity of POS tagging. To resolve this problem, we simplify the observation probability and the transition probability calculations based on the assumption that the first POS tag and the last POS tag provide important clues to syntactically connect words, as shown in

$$P_k(w_i | t_i) \approx \prod_{j=1}^{m} \left( P_k\left( \text{seg}_M_j | \text{seg}_I_j \right) \right),$$

where $k = 1 \ldots c$,

$$P(t_i | t_{i-1}) \approx P(t_i^{\text{first}} | t_{i-1}^{\text{last}}).$$

(4)
In (4), \( \text{seg}_{j} \) is the \( j \)th longest-morpheme segment that the modified LLMP model generates from the \( i \)th word, \( w_{i} \), and \( \text{seg}M_{i} \) is a morpheme sequence of the \( j \)th longest-morpheme segment. \( \text{seg}T_{j}^{\text{first}} \) and \( \text{seg}T_{j}^{\text{last}} \) are the first POS tag in the \( j \)th longest-morpheme segment and the last POS tag in the \( j-1 \)th longest-morpheme segment, respectively. \( P_{k} \) is the probability of the \( k \)th morpheme among \( c \) morpheme candidate sequences in the \( i \)th word, \( w_{i} \). \( \text{seg}T_{j}^{\text{first}} \) and \( \text{seg}T_{j}^{\text{last}} \) are a POS tag of the first morpheme in the \( i \)th word, \( w_{i} \), and a POS tag of the last morpheme in the \( i-1 \)th word, \( w_{i-1} \). Figure 6 shows an example of the simplified HMM based on the modified LLMP model.

As shown in Figure 6, the transition probability between “chong-20-nyeon-eul (for total of 20 years)” and “jip-gwon-han (doing the seizure of power)” is calculated based on grammatical possibilities between the POS tag “noun” of the first morpheme “jip-gwon” in the current word and the POS tag “postpositional_word” of the last morpheme “eul” in the previous word. The observation probability of the word “jip-gwon-han” is calculated as the maximum score among the following three probabilities:

\[
P_{1}(\text{“jip-gwon/noun” | seg_{i}}) \times P_{1}(\text{“noun” | } \emptyset) \\
\times P_{1}(\text{“han/noun” | seg_{i}}) \times P_{1}(\text{“noun” | ”noun“}), \\
P_{2}(\text{“jip-gwon/noun” | seg_{i}}) \times P_{2}(\text{“noun” | } \emptyset) \\
\times P_{2}(\text{“ha/adjective_suffix + n/ending” | seg_{i}}) \\
\times P_{2}(\text{“adjective_suffix + ending” | “noun”}), \\
P_{3}(\text{“jip-gwon/noun” | seg_{i}}) \times P_{3}(\text{“noun” | } \emptyset) \\
\times P_{3}(\text{“ha/verb_suffix + n/ending” | seg_{i}}) \\
\times P_{3}(\text{“verb_suffix + ending” | “noun”}).
\]

(5)

In the above example, we can assign \( P_{1}(\text{“jip-gwon/noun” | seg_{i}}) \times P_{1}(\text{“noun” | } \emptyset) \) to 1.0 without any calculation because the morpheme sequence of the first segment is always “jip-gwon/noun.” This strategy makes the simplified HMM use less memory. As we illustrated above through examples, the modified LLMP model ignores many morpheme candidates. Due to this pruning process, the simplified HMM can dramatically reduce the amount of calculations required to obtain the observation probabilities. Finally, the maximum scores from (1) and (4) are calculated using the well-known Viterbi algorithm [21].

4. Experiments

4.1. Data Sets and Experimental Settings. To evaluate the proposed model experimentally, we used the 21st Century Sejong Project’s POS-tagged corpus [22]. Table 2 describes the Sejong POS-tagged corpus in brief.

We divided the POS-tagged corpus into training and test data, at a ratio of nine to one. We then performed a 10-fold cross-validation using the following evaluation measures: precision, recall rate, and F1-measure. In order to evaluate the usefulness of the proposed model in a real information appliance environment, we implemented it in a commercial mobile phone with a XSCALE PXA270 CPU, 51.26 MB memory, and Windows Mobile 5.0.

4.2. Experimental Results. The first experiment performed was intended to evaluate the changes in performance with the proposed model, based on the number of decomposition rules. We computed the average performance of the proposed model at various cutoff points in Figure 7.

In Figure 7, the more rules the proposed model had, the higher performance it obtained. However, we believe that the model incorporating top-40% rules is the most suitable for information appliances because the models having more rules require more processing time and larger working memories, while delivering limited performance improvement over models with smaller rule sets.

In the second experiment, we compared the performance of the proposed model with those that are representative of previous models, using the same training and testing data, as listed in Table 3.
In Table 3, "LLMP" is a morphological analyzer based on conventional LLMP rules. This morphological analyzer does not need additional POS tagging processes because it returns one morpheme sequence per word. "Tabular parsing + HMM" is a POS tagger based on an HMM that selects the most reasonable sequence among all possible morpheme candidates generated by the tabular parsing method. The system is one of state-of-the-art Korean morphological analyzers which show F1-measures of 94–95% [18]. "Modified LLMP + Simplified HMM" is the proposed POS tagger that selects the most probable sequence among a number of morpheme candidates generated by the modified LLMP model. As listed in Table 3, "Tabular parsing + HMM" exhibited the best performance in terms of all measures. However, the performance differences between the proposed model and the "Tabular parsing + HMM" model were much smaller than those between the proposed model and the "LLMP." This fact reveals that the decomposition rules are very effective. On the other hand, the proposed model significantly outperformed the "LLMP.

In the last experiment, we compared the memory usage and response time of the above models, as listed in Table 4.

As listed in Table 4, the proposed model used much less memory and required much less processing time than the "Tabular parsing + HMM" model. Let N denote the number of eojeols in an eojeol. In the scan procedure from left to right for matching an eojeol against each key in a morpheme dictionary, the tabular parsing model has the time complexity O(N^5) because it should check all grammatical connections between two adjacent eojeols in a similar manner with CYK algorithm (http://en.wikipedia.org/wiki/CYK_algorithm). However, the modified LLMP has the time complexity O(N) as already known. Let S denote the number of observations in an HMM, and let T denote the number of transitions in an HMM. In the POS tagging procedure, the simplified HMM has the same time complexity O(TS^3) with the ordinary HMM. However, the sizes of T and S in the simplified HMM are about 3 times smaller and about 5 times smaller than those in the ordinary HMM, respectively. As shown in the time complexity analysis, "LLMP" is the most lightweight model exhibiting high speed. However, the proposed model is the more suitable for information appliances because the precision and recall rate of the "LLMP" model are too low for NLP applications. Although the computing power of information appliances is rapidly increasing, the memory usage and the processing time of the "Tabular parsing + HMM" model are still limiting factors for information appliances. For example, we can easily find many wireless sensor network (WSN) gateways and security appliances with 64–256 MBs. In many systems, the size of internal user memory is restricted within a few MBs. The experimental platform (i.e., the commercial mobile phone with a XSCALE PXA270 CPU, 51.26 MB memory, and Windows Mobile 5.0) had only 8 MBs of internal user memory which is too little to implement NLP applications. In addition, information appliances may use information retrieval (IR) techniques for extracting keywords from instruction manuals or online contents. In this case, "Modified LLMP + Simplified HMM" spent 0.3 seconds per MB for indexing keywords, but "Tabular parsing + HMM" spent 85 seconds per MB for indexing keywords. In particular, in mobile devices with restricted battery capacity, long processing times lead to rapid battery consumption. Based on these experimental results, if computational cost and memory limitations are important factors, the combination of the modified LLMP model and the simplified HMM may be one of the best solutions.

4.3. Contribution to Distributed Sensor Networks. Smart home technology can be used in the following key areas
in which various sensors should interact with each other in order to detect residents’ behaviors and protect against dangerous situations [3]:

(i) safety area: intruder detection, burglar deception, fire detection, video surveillance, and so on;

(ii) comport area: temperature control, light control, windows control, and so on.

To realize this smart home environment, sensor network systems should gather information detected by sensors and should transmit the information to tablet terminals, gateways, or information appliances. If the sensor network systems adopt NLP techniques (i.e., NLP techniques are embedded in sensors or gateways), they will be able to more promptly detect various events and more accurately determine their actions against events. For example, if a keyword detector based on NLP techniques is embedded in a motion sensor (or CCTV), the sensor network system can generate necessary actions when the keywords like “money” and “give me” are included in the conversation between an intruder and a resident or when a resident shouts “fire” while fast moving. As a result, the proposed model can contribute to making sensor network systems better at understanding residents’ contexts.

5. Conclusions

We proposed a morpheme segmentation and POS tagging model for an information appliance. To reduce the number of morpheme candidates, the proposed model uses a method that expands the set of morpheme candidates generated by longest-match-preference rules, instead of using the well-known tabular parsing method. To reduce the computational cost and memory usage, the proposed model uses a method that simplifies an inner HMM, which is necessary in order to find the correct sequence of morphemes in a word. In the experiments, the proposed model exhibited good performance in terms of the various evaluation measures such as precision, recall rate, memory usage, and response time. On the basis of these experiments, we conclude that the proposed model is suitable for information appliances with many hardware limitations because it requires less memory and consumes less battery power.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research was supported by the IT R&D program of MOTIE/MSIP/KEIT [10041678, The Original Technology Development of Interactive Intelligent Personal Assistant Software for the Information Service on multiple domains]. This research was also supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2013R1A1A4A01005074).

References

[1] M. Caillet, J. F. Pessiot, M. R. Amini, and P. Gallinari, “Unsupervised learning with term clustering for thematic segmentation of texts,” in Proceedings of the RIAO Conference (RIAO ’04), 2004.
[2] A. Kamilaris and A. Pitsillis, “Social networking of the smart home,” in Proceedings of the 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC ’08), pp. 2632–2637, September 2010.
[3] M. B. Chandak and R. Dharaskar, “Natural language processing based context sensitive, content specific architecture and its speech based implementation for smart home applications,” International Journal of Smart Home, vol. 4, no. 2, pp. 1–10, 2010.
[4] A. Hurson, Connected Computing Environment, Academic Press, 2012.
[5] B. Merialdo, “Tagging English text with a probabilistic model,” Computational Linguistics, vol. 20, no. 2, pp. 155–171, 1994.
[6] E. Brill, “Transformation-based error-driven learning and natural language processing: a case study in part-of-speech tagging,” Computational Linguistics, vol. 21, no. 4, pp. 543–564, 1995.
[7] T. Brants, “TriT: a statistical part-of-speech tagger,” in Proceedings of the Conference on Applied Natural Language Processing (ANLC ’00), pp. 224–231, Stroudsburg, Pa, USA, 2000.
[8] C. Kruengkrai, K. Uchimoto, J. Kazama, Y. Wang, K. Torisawa, and H. Isahara, “Joint Chinese Word segmentation and POS tagging using an error-driven Word-character hybrid model,” IEICE Transactions on Information and Systems, vol. E92-D, no. 12, pp. 2298–2305, 2009.
[9] M. Porter, “An algorithm for suffix stripping,” Program, vol. 14, no. 3, pp. 130–137, 1980.
[10] J. Lovins, “Development of a stemming algorithm,” Mechanical Translation and Computational Linguistics, vol. 11, no. 1-2, pp. 22–31, 1968.
[11] H. H. Htay and K. N. Murthy, “Myanmar word segmentation using syllable level longest matching,” in Proceedings of the 6th Workshop on Asian Language Resources, pp. 41–48, 2008.
[12] H. Tseng and K. Chen, “Design of Chinese morphological analyzer,” in Proceedings the 1st SIGHAN workshop on Chinese Language Processing, vol. 18, pp. 1–7, 2002.
[13] Y. Hong, M.-W. Koo, and G. Yang, “Korean morphological analyzer for speech translational system,” in Proceedings of the International Conference on Spoken Language Processing (ICSLP’96), pp. 673–676, October 1996.
[14] F. Karlsson, “Constraint grammar as a framework for parsing running text,” in Proceedings of the 13th Conference on Computational linguistics, pp. 168–173, 1990.
[15] A. Voutilainen, “A syntax-based part of speech analyzer,” in Proceedings of the 7th Conference of the European Chapter of the Association for Computational Linguistics, pp. 157–164, 1995.
[16] K. W. Church, “Stochastic parts program and noun phrase parser for unrestricted text,” in Proceedings of the Conference on Applied Natural Language Processing, pp. 136–143, 1988.
[17] E. Charniak, C. Hendrickson, N. Jacobson, and M. Perkowitz, “Equations for part-of-speech tagging,” in Proceedings of the 11th National Conference on Artificial Intelligence, pp. 784–789, July 1993.
[18] D. Lee and H. Rim, “Probabilistic models for Korean morphological analysis,” in Proceedings of the International Joint Conference on Natural Language Processing, pp. 197–202, 2005.

[19] J. Lafferty, A. McCallum, and F. Pereira, “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” in Proceedings of the International Conference on Machine Learning, pp. 282–289, 2001.

[20] T. Kudo, K. Yamamotoz, and Y. Matsumoto, “Applying conditional random fields to Japanese morphological analysis,” in Proceedings of the Conference on Empirical Methods on Natural Language Processing (EMNLP ’04), pp. 230–237, 2004.

[21] G. D. Forney Jr., “The Viterbi algorithm,” Proceedings of the IEEE, vol. 61, no. 3, pp. 268–278, 1973.

[22] “The National Institute of the Korean Language: final report on achievements of 21st Sejong project: electronic dictionary,” 2007.
