SEMANTIC UNDERSTANDING BY COMBINING EXTENDED CFG PARSER WITH HMM MODEL

Yushi Xu[1], *Stephanie Seneff*[1], *Alice Li*[1], *Joe Polifroni*[2]

[1] Spoken Language Systems Group, MIT CSAIL, 32 Vassar Street, Cambridge MA, USA
[2] Nokia Research Center, 4 Cambridge Center, Cambridge MA, USA

ABSTRACT

This paper presents a method for extracting both syntactic and semantic tags. An extended CFG parser works in conjunction with an HMM model, which handles unknown words and partially known words, to yield a complete syntactic and semantic interpretation of the utterance. Four experiments and applications were performed using the paradigm to show the usefulness of the approach in processing spoken sentences.

Index Terms— HMM, Lexicon Development, Data Labeling, Spoken Language Processing

1. INTRODUCTION

The task of semantic understanding in the context of spoken dialogue systems is a well-researched topic [1] [2] [3]. While methods based on explicit parsing often provide a more reliable interpretation than purely stochastic methods, they are prone to hard failure, and this may be especially problematic when faced with errorful recognition outputs. On the other hand, parse failure can be useful as a rejection criterion, protecting the system from providing an inappropriate response to an out-of-domain utterance.

Recently, machine-learning methods have gained popularity for the task of extracting key semantic information from utterances, while essentially bypassing syntactic analysis. The field of named-entity extraction, which is closely related to speech understanding for spoken dialogue systems, is replete with examples of approaches that utilize a set of carefully chosen features and a machine learning algorithm to model the mapping from features to semantic tags [4].

We have been developing spoken dialogue systems for many years, and our approach utilizes a syntax-based lexicalized context-free grammar (CFG), which depends on explicit knowledge of the vocabulary, but can utilize semantic tags to allow unknown words to parse under specially designated categories such as "restaurant_name" or "street_name" [5]. Such a grammar is designed to work with a dynamic recognizer that emits these tags automatically, and a mechanism exists in our dialogue systems to populate these special categories on the fly in the recognizer with explicit vocabulary items based on dialogue context. However, typed sentences would not contain such tags, and hence a research topic arises which involves automatically tagging typed queries entered into the system.

Amazon Mechanical Turk (AMT) provides a new opportunity to collect inexpensive and reasonably high quality data, to use for system development in spoken dialogue systems. Such data can be essential for building new natural language applications or adapting an existing application to a related domain. The data help define the domain vocabulary and can be used as training data for language modeling and system development. However, such data typically come without any semantic tags, so techniques to automatically acquire such tags are needed, unless expertise is devoted to this task, which defeats the purpose of cheap data collection.

Even in a domain without explicit tags, it is difficult to acquire full knowledge of all vocabulary items, particularly in the large or even open classes such as adjective, noun, and proper noun. Our parser supports unknown words in noun classes. However, some words are known to the parser, but not known under the appropriate syntactic category in the sentence. For example, if "win" were only known as a verb, the sentence, "talking to him was a big win" would not parse. However, if we allow both known and unknown words to be supported under "noun," then the parse space grows astronomically, which is problematic. We refer to this problem as "category mutation." Expanding the grammar to improve coverage on a new set of utterances for a new domain is typically labor intensive and requires expertise.

In this paper, we propose a novel method for extracting both syntactic and semantic tags, and then utilizing that information to aid in the process of acquiring a semantic understanding component for a new domain. The system includes an explicit model to handle both unknown words and words that are only partially known (as in the example "win" above). The extended CFG parser works in conjunction with an HMM model to yield a complete syntactic and semantic interpretation of the utterance, even in the presence of unknown or partially known words. The system automatically proposes a set of new grammar rules to associate words with preterminal categories, which are then subject to approval by the developer.

Several applications using the model will be discussed, involving grammar lexicon development and data labeling for dynamic language models in the speech recognizer. In the first case, the data apply to a new domain, and typically a large number of unknown words are present. The purpose is to generate a new lexicon for a CFG grammar to account for proper category assignment for the new vocabulary in the data. In the second case, the goal is to tag out the phrases that belong to the pre-defined dynamic classes of a dynamic language model. In both cases, proper handling of unknown words and category mutation is the key focus.

The rest of the paper is organized as follows: Section 2 will give a brief discussion of HMM-based data labeling models. Section 3 will introduce our approach. Section 4 will show the application of the model in terms of lexicon development and dynamic class labeling. Section 5 will conclude the paper and point to some future work.

2. HMM-BASED LABELING

Two common word-level labeling tasks are part-of-speech tagging and named entity recognition. Although the tags differ for the two cases, both can be described as follows: given an input word sequence w_1, …, w_n, output a sequence of tags or labels t_1, … t_n. The labels are usually syntactically or semantically meaningful, such as the part-of-speech or the named entity class of the word.

Hidden Markov Models are commonly used to solve this problem, as described in [6] and [7]. In this generative view, the word sequence is the observation sequence, and the labels are the hidden states. The next word w_i is generated by first generating its label t_i based on a finite history of previous labels t_{i-1}, t_{i-2}, …. Then w_i is generated according to the emission probability, which can be modeled as either only depending on the current state t_i, or depending on both the current state and the state history.

One important problem in this model is how to deal with the unknown words, which are especially problematic when the training data are limited and inconsistent with the test data. Several methods to handle unknown words have been proposed. Word features are commonly used in computing emission probabilities for unknown words. For English, features such as capitalization and suffixes provide effective information for part-of-speech tagging [6] [8]. For languages like Chinese and Japanese, which do not have rich morphology, character level information is used to help guess the correct label [9] [10] [11]. Other methods that are not based on word analysis include the Hapax Legomena approach, which assumes the distribution of labels over unknown words is similar to the distribution of labels over words that occur only once in the training data [12]. Wang and Acero used a combination of CFG and HMM to learn a semantic grammar from a semantic scheme, a grammar library and labeled training data. An HMM model is incorporated to encounter problems in slot-poor low resolution tasks and robustness [13].

3. MODEL

3.1. Extended CFG Parser

The labeling approach we use integrates a CFG parser with an HMM preterminal model. The parser [13] uses a CFG grammar augmented with a set of non-context-free constraints to perform parsing. The grammar includes both syntactic patterns and the corresponding lexicon. A statistical model, automatically trained on parsed sentences, captures the spatial-temporal probabilities of the parse tree nodes, allowing it to select the best-scoring candidate among ambiguous parses. The parser is fast, small-foot-print, and robust to spoken language characteristics.

The labeling system takes in an input sentence (pre-segmented in the case of Chinese), parses it, and then outputs a sequence of labels which are the immediate parents of each terminal word in the parse tree, *i.e.* preterminals. The preterminals can be syntactic or semantic categories. Thus, the system can perform various types of labeling, including but not limited to part-of-speech tagging and named entity recognition.

Three situations can cause parse failure: maximum stack space exceeded, unknown words and uncovered grammar patterns. The last situation, which indicates an under-developed grammar, is outside the scope of this paper. In the other two cases, a back-up HMM model, discussed in more detail in the next subsection, is used to first hypothesize N-best preterminal sequences. It includes an unknown word model, which hypothesizes categories for words that do not exist in the lexicon, as well as hypothesize new categories for known words. The parser then uses the restrictions of the preterminal sequence hypotheses one by one, and selects the output that has the highest probability, combining the HMM preterminal sequence score and the parse tree score.

There are three major advantages in integrating labeling with parsing. First, with the actual parsing process, the output labels are restricted by the grammar. Long distance dependency can be captured and grammatically inappropriate label sequences are avoided. Second, although vocabulary varies largely from domain to domain, the syntax of a language rarely changes among different domains. Thus, a generic grammar trained with a corpus in a broad coverage domain can be used to label data in other domains. Lastly, the system's performance as a parser improves with the integration of the HMM preterminal model. Sentences containing unknown words and/or words with category mutation, and sentences that take up the parser's stack space are likely to be parsed after preterminal sequences are hypothesized. The parsing process is very fast, given the restrictions on the preterminal categories.

3.2. HMM Model

The HMM model is defined using the following equation.

$$\begin{aligned}&\underset{t_1,\dots,t_n}{\operatorname{argmax}} P(t_1,\dots,t_n|w_1,\dots,w_n)\\ &= \underset{t_1,\dots,t_n}{\operatorname{argmax}} P(w_1,\dots,w_n|t_1,\dots,t_n)P(t_1,\dots,t_n)\end{aligned} \tag{1}$$

$$= \underset{t_1,\dots,t_n}{\operatorname{argmax}} \left[\prod_i P(w_i|t_i) \right] P(t_1,\dots,t_n)$$

We assume that the generation of a word only depends on the current preterminal. Thus, the computation of the best preterminal (label) sequence $t_1, \dots, t_n$ given the word sequence $w_1, \dots, w_n$ can be decomposed into two parts: $P(t_1,\dots,t_n)$ and $P(w_i|t_i)$. An N-gram model encodes $P(t_1,\dots,t_n)$, the preterminal sequence language model. $P(w_i|t_i)$, the emission probability, can be calculated using Equation (2).

$$P(w_i|t_i) = \frac{c(w_i,t_i)+1}{c(t_i)+\mathrm{size}(t_i)} \quad (2)$$

$c(\cdot)$ denotes the empirical count of a specific word or preterminal. Size$(\cdot)$ is the number of words defined under the preterminal category. Thus, each word defined under the preterminal category has a minimal count of 1.

3.3. Unknown Words and Category Mutation

When a word is not defined in the grammar, or has not been defined under the desired preterminal category (category mutation), the unknown model is used instead of the normal emission probability. In the grammar, a special notation is used to indicate preterminal categories that are allowed to accept unknown words, such as nouns, proper nouns, *etc.* These preterminal categories are selected by the developers according to the specific application. We will refer to these categories as "open categories" and denote them as t^o in the rest of the paper. The probability of an unknown word given category t^o is calculated using the following two criteria.

$$P_{unk}(w|t^o) \le \frac{1}{c(t^o)+\mathrm{size}(t^o)} \quad (3)$$

$$P_{unk}(w|t^o) \propto c(t^o)+\mathrm{size}(t^o) \quad (4)$$

P_{unk} applies to all word that do not exist under category t^o, *i.e.*, regardless of whether the word is defined under other categories. The first criterion ensures that the unknown probability is less than that of any known word under the category. The second criterion states that, the more known words an open category has, the more likely it is to have an unknown word. Assuming an open category will never become closed, and the size always grows at a constant rate, then a large-vocabulary category can be viewed as the result of constant creation of new words under the category, and so it is more likely to accept new words. The number of occurrences accounts for those categories that do not have many words, but yet are highly likely to have new words because of frequent occurrence, such as the measure words in Chinese. Note that this criterion would not cause a problem for frequent categories such as English articles, which should not be open categories.

After the two criteria are satisfied, emission probabilities for the known words are renormalized to ensure that the probabilities of an open category sum to one.

3.4. Unknown Character Model

In addition to the unknown word model, an alternative unknown character model can be used to calculate the unknown word probability for Chinese. The meanings and positions of the characters in a word contribute to the meaning of the whole word. Furthermore, different categories tend to have different distributions over number of characters in the word. Thus, it is useful to calculate $p(unk|t^o)$ on the basis of characters.

The unknown character model can be expressed using the following equation.

$$P_{unk}(w|t^o) = P_{len}(len_w|t^o) \cdot P_{init}(c_1|t^o) \cdot P_{end}(c_n|t^o) \cdot \prod_i P(c_i|t^o) \quad (5)$$

In the equation, len_w indicates the number of characters in the unknown word. $P_{init}(c_1|t^o)$ and $P_{end}(c_n|t^o)$ indicate, given category t^o, the probabilities of starting with character c_1 and ending with character c_n, respectively. These two terms take advantage of the position information of the starting and ending character. Term $\prod_i P(c_i|t^o)$ accounts for the probabilities of each character given category t^o in without regard to sequence order.

If a character in the word never occurs in category t^o, its probability is calculated using the same criteria as Equations (3) and (4), except that $c(t^o)$ and $\mathrm{size}(t^o)$ are calculated on the character base instead of on the word base.

For languages like English, something similar to the unknown character model could be constructed based on prefixes and suffixes.

3.5. Implementation

The HMM model together with the unknown word model is directly implemented using Finite State Transducers (FSTs) [14]. The N-best preterminal sequences are obtained by composing the following three FSTs and outputting N-best paths using a standard FST search algorithm.

- Input FST: maps each word in the sentence to itself. If the word does not exist in the grammar, maps it to a special unknown symbol.
- Word-Preterminal FST: maps each word to its defined categories with $P(w|t)$. Also maps each word, including the unknown symbol, to all the open categories with $P_{unk}(w|t^o)$.
- Preterminal N-gram FST: contains the N-gram language model for the preterminals.

In addition, P_{unk} for out-of-vocabulary words is computed under the unknown character model for Chinese.

4. APPLICATIONS

4.1. Lexicon Improvement in Generic Chinese Domain

The first experimental application was carried out in a Chinese travel domain. A generic grammar had been previously developed. However, the grammar is under-developed, *i.e.*, it does not cover all the syntactic patterns or the entire vocabulary in the domain. The experiments we carried out were aimed at enlarging the lexicon by using the extended CFG parser to automatically choose categories for the unknown words.

The corpus we used for the experiments was IWSLT05, which consists of 20,000 sentences in the spoken travel domain. The sentences were pre-segmented into words, and chunked according to periods before carrying out the experiments. After chunking, altogether 23,768 sentences were produced. The unique vocabulary size is 8,187, and 8.8% (727) were not included in the grammar.

The grammar contains 282 preterminal categories. Among those, eleven were defined as open categories: noun, adjective, verb, generic adverb, noun measure word, verb measure word, unit, front conjunction, back conjunction, location particle and uncommon preposition. Since there are no manually labeled training data to train the unknown word model, we use the following unsupervised procedure to perform lexicon improvement.

a) Parse the whole corpus without the back-up HMM model.
b) Tag the words for all the parsable sentences with their preterminals according to the top parse tree.
c) Create the HMM model and the N-gram preterminal language model using the tagged data from (b).
d) Parse the unparsable sentences with the HMM model obtained in Step (c). Enlarge the lexicon based on the new preterminal-terminal pairs in the parsed sentences.

After Step (a), 19,159 (80.6%) sentences parsed, and were used to build the FSTs. The resulting 4-gram preterminal language model is about the same size as a trigram LM built from the terminal words. The remaining 4,609 failed sentences were used in Step (d) to produce the new lexicon. Table 1 shows the statistics of the newly obtained lexicon using difference settings. The preterminal category assignment accuracy was judged manually. Since the grammar was underdeveloped, parse failure in Step (d) could be caused either by incorrect preterminal assignment or by uncovered syntactic patterns. To distinguish between the two factors, an oracle model was created for comparison. It assigns each unknown word to the most appropriate preterminal category among the eleven open categories. This gives an upper bound for the extracted unknown lexicon under the imperfect grammar.

The first three models gave similar numbers of correct preterminal assignments for unknown words. The accuracy, however, increases from 0.64 to 0.8 with the word length feature and character-level model. Using 5-best preterminal hypotheses increased recall but sacrificed accuracy.

Since the training of the model does not rely on manual labeling, the approach can be used iteratively. After a new lexicon is produced, a human expert judged the correctness of the preterminal assignments, added the correct ones into the original grammar, excluded the incorrect ones from the HMM model, and repeated the process. By doing so, more sentences will be parsed and serve as training data in the next iteration. We performed three iterations using the 1-best Char+Len model. The final results are shown in the fifth line of Table 1. The recall is close to that of the 5-best Char+Len model, but the accuracy remains high. After three iteration, the parse coverage increased from 80.6% to 86.8%.

	Unknown Words			Category Mutation		
	# total	# correct	accuracy	# total	# correct	accuracy
Word	487	320	0.64	899	279	0.31
Word +Len	480	341	0.71	668	226	0.34
Char +Len	405	322	0.80	74	52	0.70
Char +Len 5-best	559	385	0.69	229	116	0.50
Char +Len 3-iters	486	377	0.77	93	60	0.65
Oracle	470	470	1	-	-	-

Table 1. Statistics of newly obtained lexicon: total numbers of terminals generated, numbers of terminals with correct preterminal assignment, and accuracy.

For the category mutation, it is observed that moving from word-level model to the character-level model significantly reduced the size of the generated lexicon. The training data contains the information of the word with the original preterminal assignment, and by using the information contained in each of the characters, the character-level model sets up a stronger relationship between the word and its original preterminal assignment. Thus, it is harder for a word to map to preterminal categories besides the ones already defined in the grammar.

Table 2 shows the accuracy of each open category of the first iteration using the 1-best length feature and character-level model. All categories had fairly high accuracy except adjectives. Word formation of adjectives in Chinese is very flexible. Also, nouns, verbs and other words can also serve as modifiers, making it difficult to distinguish them through syntactic structure.

noun	187/227 (82%)	adjective	20/60 (33%)	verb	149/173 (86%)
adverb	3/3 (100%)	n-measure	7/7 (100%)	v-measure	0
unit	1/1 (100%)	f-conj	2/3 (67%)	b-conj	1/1 (100%)
lc-particle	3/3 (100%)	uncommon preposition	2/2 (100%)		

Table 2. Category-wise accuracy of the first iteration using the length feature and character-level model (including both unknown words and category mutation).

4.2 Tagging Named Entities in the Restaurant Domain

An interesting application for the parsing resource described in this paper is to automatically tag data obtained via typed inputs, to be used for training the statistics of a spoken dialogue system. We have developed a system that provides information about restaurants, hotels, and other landmarks in a major metropolitan area [15], and we recently collected over 13,000 typed queries from AMT. We asked workers to imagine a system that could answer such questions and then to make up appropriate queries for such a system. Our recognizer is based on a dynamic framework where the contents of tagged entities are populated via the appropriate database at any time in the conversation, so the actual names provided by Turkers are not important to us. However, they need to be tagged, among the categories, *street name, neighborhood, city, landmark, restaurant, hotel, museum,* and *subway*, before we can use them either to train the language model for our recognizer or to process them through our natural language parsing component to verify proper understanding.

how do I get from <streetname> main </streetname> and <streetname> chicago </streetname> in <city> evanston </city> to <landmark> century theater </landmark>?
what's the phone number of <restaurant> rudyard's pub </restaurant> in <neighborhood> houston </neighborhood>
where is the <restaurant> best ice cream shop </restaurant> near <landmark> central park </landmark> in <city> new york city </city>?
what is the address for the <landmark> cleveland indians baseball stadium </landmark>?
driving directions from <landmark> hunter's point </landmark> to <landmark> sf general hospital er </landmark>
what family friendly restaurants are within walking distance of <landmark> independence hall </landmark> in <city> philadelphia </city>

Figure 1. Examples of automatically tagged sentences from the city guide domain.

We approached this problem by building an FST mapping an untagged sentence to an output sequence that encodes the preterminal layers of the parse trees, including the tags, which are embedded in the preterminal labels of the taggable entities. We trained it on a large corpus of synthesized utterances obtained by generalizing from a previously collected data set. An N-best list (N=10) of candidates was then processed by our parser, constrained by the strong filter provided by the N-best preterminal sequence hypotheses. In order to obtain a variety of options for the tags, we randomly chose among the top three candidates when more than one hypothesis could parse.

Figure 1 shows examples of several tagged sentences that were obtained using this strategy. Of course there are many errors in the tags, but for the most part they don't really detract from our ability to make good use of the utterances. For example, "best ice cream shop" is obviously not a restaurant name, but the sentence is a perfectly fine example of a plausible syntactic construct. Since our language model discards the contents of all tagged entities, replacing them with the tag identifier, there is no negative consequence of mistagging, as long as the resulting sentence is well formed and semantically plausible. The strong filter based on grammatical constraint usually reinforces well-formedness. We used these data to expand grammar coverage, ultimately obtaining a parse coverage of 94.2% on the data set.

4.3. Tagging Adjectives and Nouns in a Restaurant Review Domain

We have spent considerable research effort recently developing a system that can process consumer-provided restaurant reviews to obtain both ratings along dimensions such as food and service as well as a set of informative descriptive phrases, such as "superb margaritas," to serve as succinct descriptions of the restaurant's special features [16]. Our methods are based on a parse-and-paraphrase paradigm, where we extract key [adverb]-adjective-noun patterns from appropriate syntactic units. Until now, we have approached this problem by explicitly lexicalizing all the significant nouns and adjectives that show up in reviews, which is a time-consuming process.

Based on the success of our algorithm for tagging named entities, we are now exploring an alternative strategy for parsing review sentences which involves automatically tagging all the nouns and adjectives, thus not requiring the grammar to have explicit knowledge of the vocabulary contained in these large classes of words. We were able to train the model mapping from words to preterminal categories (including the tag labels) by taking advantage of an existing grammar which had explicit knowledge of the noun and adjective lexica. We utilized a very simple machine-learning strategy to map every word in the sentence to three categories with associated probabilities: adjective, noun, or other. Our statistical model utilizes two sets of features, one based on word unigram probabilities, and the other based on letter sequence trigrams in the words in each of the three categories. These statistics were trained on a large corpus of parsed utterances obtained from the Web. The parser selected from an N-best list (N=10) of tagged candidates. We assessed the effectiveness of the strategy by comparing the descriptive phrases extracted from the grammars with and without explicit lexical knowledge.

Evaluation was performed on the transcripts of a set of spoken utterances that were acquired by having subjects describe orally their opinions about a restaurant they had recently dined at. We manually compared extracted descriptive phrases obtained via the procedures outlined in [16] obtained via the two methods (tagged versus untagged sentences). For the untagged sentences, we used a grammar that could only parse a word as a noun or an adjective if it was marked with the appropriate tag.

In total, 173 phrases were extracted from the tagged sentences that were not extracted from the untagged sentences. By manual inspection, we judged 119 of these to be useful as descriptors, such as "amazing steak," "attentive waitress," or "crowded bar area." The reason for its higher yield is that it realizes significantly better parse coverage, because it does not have to explicitly know the nouns and adjectives of the domain. This also means that the grammar can remain static, while the encoding of nouns and adjectives becomes a separate task carried out by the machine-learning algorithm.

4.4. Experiments in a Medical Domain

Our final experiment involves developing a grammar for a new domain. We envision a system that allows users to access a database of thousands of comments provided via a chat-room forum, on the specific subject of statin drug side effects. This domain is very different from our previous domains -- many words are out of vocabulary and the grammar constructs are often more complex. We have created a preliminary version of the Web page, and have obtained nearly 1,000 queries via AMT by presenting workers with extracted comments and asking them to come up with a query for which the comment would be an appropriate hit in a web search based on the query. We used our new tool to automatically acquire proposed preterminal-to-terminal mappings for novel words appearing in the queries. An expert developer then manually repaired the automatically obtained grammar rules, and spent a few hours expanding the grammar rules to reflect new patterns in the sentences. A total of 588 new word-preterminal mappings were created. After this grammar development phase, about 91% of the sentences could be parsed by the new grammar. Example sentences are shown in Figure 2.

Is my thinning hair due to the statins I'm on?
Do statin drugs intended to lower cholesterol lead to diseases like ALS?
Can the liver tests that are done when one has high cholesterol remain abnormal for years?
Could Zocor be the cause of shoulder and neck pain in an otherwise healthy forty one year old female?
Can pharmacists tell the patients that they should take coenzyme q ten for pain relief while taking statins?
Could the joint pain that I have been dealing with lately be a side effect of the statins I have been prescribed?

Figure 2. Example sentences from the drug side-effect domain that obtain a full parse analysis.

5. CONCLUSIONS

We have presented a framework for speech understanding which utilizes an extended CFG parser in conjunction with an HMM model to handle unknown words and partially known words. The HMM model is able to exploit character/morpheme level information to assign preterminal categories for each word in the sentence, and then restrict the parsing output based on the preterminal sequence. The approach takes advantage of the knowledge presented in an existing grammar, and thus does not require labeled training data. We have applied the approach in four different tasks of lexicon development and data labeling in different domains. In the future, we would like to extend the model to handle sentences that fail due to uncovered syntactic patterns.

6. REFERENCES

[1] R. De Mori et al., "SPOKEN LANGUAGE UNDERSTANDING: A SURVEY," *IEEE Signal Processing magazine*, vol. 25, no. 3, pp. 50-58, 2008.

[2] S. Seneff, "Response Planning and Generation in the Mercury Flight Reservation System," *Computer Speech and Language*, vol. 16, pp. 283-312, 2002.

[3] L. S. Zettlemoyer and M. Collins, "Learning Context-dependent Mappings from Sentences to Logical Form," in *Proc. ACL-IJCNLP*, Singapore, 2009.

[4] D. Nadeau and S. Sekine, "A survey of named entity recognition and classification," *Linguisticae Investigationes*, vol. 30, no. 1, pp. 3-26, 2007.

[5] G. Chung, S. Seneff, W. Chao, and L. Hetherington, "A Dynamic Vocabulary Spoken Dialogue Interface," in *Proc. Interspeech*, Jeju, South Korea, 2004, pp. 327-330.

[6] T. Brants, "TnT -- A Statistical Part-of-Speech Tagger," in *Proc. the sixth conference on Applied natural language processing*, Seattle, WA, USA, 2000, pp. 224-231.

[7] D. M. Bikel, S. Miller, R. Schwartz, and R. Weischedel, "Nymble: a high-performance learning name-finder," in *Proc. the Fifth Conference on Applied Natural Language Processing*, Washington DC, USA, 1997, pp. 194-201.

[8] R. Weischedel, R. Schwartz, J. Palmucci, M. Meteer, and L. Ramshaw, "Coping with ambiguity and unknown words through probabilistic models," *Computational Linguistics*, vol. 19, no. 2, pp. 361-382, 1993.

[9] Z. Huang, M. P. Harper, and W. Wang, "Mandarin Part-of-Speech Tagging and Discriminative Reranking," *Proc. EMNLP 2007*, 2007.

[10] T. Nakagawa and K. Uchimoto, "A Hybrid Approach to Word Segmentation and POS Tagging," in *Proc. the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, Prague, Czech Republic, 2007, pp. 217-220.

[11] C. Kruengkrai et al., "An Error-DrivenWord-Character Hybrid Model for Joint Chinese Word Segmentation and POS Tagging," in *Proc. AFNLP (ACL-IJCNLP)*, Suntec, Singapore, 2009, pp. 513-521.

[12] D. Jurafsky and M. H. James, *Speech and Language Processing, An Introduction to Natural Language Processing,Computational Linguistics, and Speech Recognition*, 2nd ed.: Prentice-Hall, 2008.

[13] Y.-Y. Wang and A. Acero, "Combination of CFG and N-gram Modeling in Semantic Grammar Learning," in *Proc. Eurospeech*, Geneva, Switzerland, 2003, pp. 2809-2812.

[14] S. Seneff, "TINA: A Natural Language System for Spoken Language Applications," *Computer Linguistics*, vol. 18, no. 1, pp. 61-86, March 1992.

[15] M. Mohri, "Finite-state transducers in language and speech processing," *Computational Linguistics*, vol. 23, no. 2, pp. 269-312, June 1997.

[16] A. Gruenstein and S. Seneff, "Releasing a Multimodal Dialogue System into the Wild: User Support Mechanisms," in *Proc. SIGDial 2007*, Belgium, 2007, pp. 111-119.

[17] J. Liu and S. Seneff, "Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm," in *Proc. EMNLP*, Singapore, 2009, pp. 161-169.