New Language Models Using Phrase Structures Extracted From Parse Trees

Takatoshi Jitsuhiro, Hirofumi Yamamoto, Setsuo Yamada; Yoshinori Sagisaka

ATR Spoken Language Translation Research Laboratories
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0288 Japan
{takatoshi.jitsuhiro, hirofumi.yamamoto, yoshinori.sagisaka}@slt.atr.co.jp

Abstract

This paper proposes a new speech recognition scheme using three linguistic constraints. Multi-class composite bigram models [1] are used in the first and second passes to reflect word-neighboring characteristics as an extension of conventional word n-gram models. Trigram models with constituent boundary markers and word pattern models are both used in the third pass to utilize phrasal constraints and headword co-occurrences, respectively. These two models are made using a training text corpus with phrase structures given by an example-based Transfer-Driven Machine Translation (TDMT) parser [2]. Speech recognition experiments show that the new recognition scheme reduces word errors 9.50% from the conventional scheme by using word-neighboring characteristics, that is only the multi-class composite bigram models.

1. Introduction

In large vocabulary continuous speech recognition, word n-gram models are widely and effectively used as language models. However, they can represent only local constraints within a few successive words and lack the ability to capture the global structures of sentences. Trigger models have been proposed to cope with these weaknesses [3]. They can model word co-occurrence characteristics. S. Zhang et al. also proposed a solution called Linkgram [4]. This model has word pairs extracted from parse trees, and it can represent syntactic relations between word pairs. The constraints, however, are weak for the global structures of sentences. Chelba and Jelinek proposed a method using a parser [5]. They used a stochastic parser itself as a language model. This language model has powerful syntactic constraints, but requires a lot of calculations.

To represent phrasal structures, we propose two kinds of language models using a parser. The first one is an n-gram model including phrase-level constraints, and the second one is a word pattern model extracted from parse trees. The parser that we use is a constituent boundary parser developed for example-based machine translation at ATR. It is called Transfer-Driven Machine Translation (TDMT). With TDMT, a sentence is analyzed into content words and boundary words to represent the sentence structure. The boundary words include both function words and markers inserted between successive content words. The boundary words represent relations among content words. We create the n-gram models with boundary markers from these analyzed sentences to utilize phrasal constraints. Furthermore, we use word patterns extracted from parse trees. The word pattern models include not only phrasal constraints but also neighboring phrasal constraints, which appear as syntactic long-distant constraints between words. Because words can represent syntactic relations between neighboring phrases, and word patterns include headwords, the combination of these language models can represent phrasal structured constraints, and word pattern models are better than trigger models because a word pattern has more than two words, and includes much more syntactic structures. These features are especially suitable for spontaneous speech. Furthermore, these language models do not require a lot of calculations because analyzing boundary marker words does not require a lot of calculations, and parsing is not needed to use the word pattern models for recognition.

In this paper, we describe these new language models in Section 2. In Section 3, we apply these new models for the task of speech recognition of ATR travel dialogues in Japanese.

2. New language models reflected by phrasal constraints

We propose new language models to make phrasal constraints. First, we describe training and recognition schemes. Second, we describe the constituent boundary parsing to obtain phrase-level structures. Furthermore, we explain how to extract word patterns from parse trees to obtain phrase structures and headword co-occurrences, and define the probabilities of the new language models.

2.1. Training and recognition schemes

Figure 1 shows the proposed training and recognition schemes. Three types of language models are made in the training scheme. First, multi-class composite bigram models are made from the original training database. These models are used for the first and second passes in the recognition process. Second, boundary markers are inserted into the sentences. In addition, some words in the sentences are combined, or divided for translation convenience. Following these steps, trigram models are made from this modified training database. Obviously, these trigram models include boundary markers. Furthermore, parse trees are made by the parser, and word pattern models are extracted from these parse trees.

In the recognition process, before the third pass search, boundary markers are inserted into the N-best candidates or a word lattice from the second pass. In the third pass search, these inputs are rescored by both trigram models with boundary markers and word pattern models. The parser is not needed to apply word pattern models into the candidates. The decoder only searches for word pattern models for some word patterns included in a target word history. Then, matched word patterns are used for rescoring.

*Setsuo Yamada is currently at NTT CS Labs., 2-4, Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan. E-mail: syamada@cslab.kecl.ntt.co.jp
2.2. Constituent boundary parsing

As a parser of spoken-language translation, ATR proposed a method called constituent boundary parsing which uses pattern matching on the surface form [2]. Constituent boundary parsing consistently describes the syntactic structures of various expressions with surface patterns consisting of variables and constituent boundary words. These variables are content words and phrases. The constituent boundary words are function words and part-of-speech bigram markers. To parse sentences by analyzing only surface patterns, part-of-speech bigram markers are inserted into content words by using part-of-speech tags and some rules before the parsing. In this paper, we call part-of-speech bigram markers as boundary markers.

These boundary words can represent relations among words or phrases. To represent phrase-level constraints, we use these analyzed sentences to make n-gram models. Furthermore, this parsing algorithm makes sentence structures using simple word patterns with boundary words consisting of function words and boundary markers. Each pattern represents a part of a sentence structure, for example, a phrase level structure. Therefore, these word patterns can represent syntactic constraints more directly than conventional n-gram models, and we use these word patterns as a language model.

We describe an example for parsing in English. For instance, the expression “go to Kyoto” in English is divided into two constituents “go” and “Kyoto,” and the preposition “to” can be identified as a constituent boundary word. Therefore, in parsing “go to Kyoto,” the pattern “X to Y” is used, which has two variables X and Y, and a constituent boundary “to.” The expression “I go” can be divided into two constituents “I” and “go.” However, it has no surface word that divides the expression into two constituents. In this case, a boundary marker is introduced as a constituent boundary. “I” and “go” are a pronoun and a verb, respectively, so the marker “pronoun-verb” is inserted as a boundary marker, “I <pronoun-verb> go.” Accordingly, “I go” comes to have the structure in Figure 2(a), and “I go to Kyoto” comes to have the structure in Figure 2(b).

This method is language independent and can be applied to Japanese. In Japanese, function words are often omitted, especially in the case of spontaneous speech. To deal with such cases, boundary markers are inserted into places where function words are omitted. Therefore, the parser can make structures by analyzing only surface patterns. Accordingly, this parsing method is effective for spoken languages.

For example, Figure 3(a) shows a parse tree of “watashi ha Kyoto he iku,” which means “I go to Kyoto” in English. In Japanese spontaneous speech, this sentence may become “watashi, Kyoto he iku.” The function word “ha” is removed in this case. In Japanese, function words are sometimes omitted, especially for spontaneous speech. To cope with such problems, the parser first inserts a boundary marker between “watashi” and “Kyoto,” like “watashi <pronoun-verb> Kyoto he iku.” Second, it extracts a structure like that in Figure 3(b). Therefore, the same structure as (a) is obtained even if some function words are omitted.

This parser also converts words by, for example, combination, division, or normalization.

2.3. Extraction of word patterns

Each sub-tree in these parse trees includes content words, function words, markers, and headwords. We define the words in a sub-tree as a set of word pattern features. Figure 4 shows how word patterns are extracted for an example sentence in English. A level 1 word pattern is defined as a set of words included in one sub-tree. In this figure, the upper sub-tree has three words as “I” ∼ “<pronoun-verb>,” and “go.” The probability of this word set is defined as $P(\text{go}|\text{I}, <\text{pronoun-verb}>).$ Because
the word order is kept, this word set keeps the sub-tree structure as is, and has a relation for the neighboring phrases by including the headword “go.” A level 2 word pattern is defined as a set of words included in two contiguous sub-trees. The probability is defined as \( P(Kyoto \mid I, <\text{pronoun-verb}>, \text{go}, \text{to}) \) in this example. Level 2 patterns have longer word sequences than level 1 patterns. Therefore, pattern models at level 2 have more constraints than at level 1. The number of words included in a word pattern model is variable; an average of two or three words for a level 1 pattern, five or six words for a level 2 pattern. In addition, both level 1 and level 2 patterns can be merged, and can be used at once.

### 2.4. Definitions of proposed language models

The probability of a word pattern model is defined as an n-gram probability,

\[
P(h_i \mid h_{-N+1}, \ldots, h_{-1}, h_{1}, \ldots, h_{N}) = \frac{C(h_i, h_{-N+1}, \ldots, h_{-1}, h_{1}, \ldots, h_{N})}{\sum_{h_i} C(h_{-N+1}, \ldots, h_{-1}, h_{1}, \ldots, h_{N})}
\]

where \( C(h_i, h_{-N+1}, \ldots, h_{-1}, h_{1}, \ldots, h_{N}) \) is the frequency of word pattern \( \{h_{-N+1}, \ldots, h_{-1}, h_{1}, \ldots, h_{N}\} \) with \( N \) words in the training data. These words are not necessarily contiguous. The word \( h_i \) means any kind of word, e.g., a content word, a headword, a function word, or a boundary marker. The numerator is the frequency of the target pattern, and the denominator is the frequency of patterns having the same left context as the target pattern.

One word in a sentence usually has some word patterns included in the word pattern models extracted from the training data. We define the pattern probability for the current word \( w_i \) as follows:

\[
P_{\text{pattern}}(w_i \mid W_{1:i-1}^{-1}) = \frac{1}{N_p} \sum_{w_p} P_{\text{pattern}}(w_i \mid W_{1:i-1}^{-1})
\]

where \( w_p \) denotes a word pattern included in the word history, and \( W_{1:i-1}^{-1} = \{w_1, w_2, \ldots, w_{i-1}\} \). \( N_p \) is the number of word patterns for the current word. This equation means an average of probabilities.

Next, we define a probability combined with an n-gram probability with boundary markers and a word pattern probability.

\[
P(w_i \mid W_{1:i-1}^{-1}) = \lambda \cdot P_{\text{pattern}}(w_i \mid W_{1:i-1}^{-1}) + (1 - \lambda) \cdot P_{\text{trigram}}(w_i \mid w_{i-2}, w_{i-1})
\]

where \( P_{\text{pattern}} \) is a probability of word patterns defined in Equation (2), \( P_{\text{trigram}} \) is a trigram probability with boundary markers, and \( \lambda \) is a constant. This is an ordinary definition by interpolation.

### 3. Experiments

#### 3.1. Experimental conditions

Experiments were carried out using the proposed language models for the Japanese travel dialogues in the ATR spontaneous speech database [6]. For the acoustic models, gender-dependent HMMs with 1,403 states and ten Gaussian mixtures per state were made by the ML-SSS algorithm [7]. For the language training set, we used 7,195 one-side dialogues which included \( 1.6 \times 10^4 \) words. The vocabulary size in the set was 16,355. For the test set, we used one side of 42 dialogues with 5,880 words that were not included in the training set.

After boundary markers were inserted and some words were modified by parsing, the training set had \( 1.2 \times 10^5 \) words, and the test set had 5,353 words that included 676 boundary markers. The number of different kinds of boundary markers was 98. Using the parsed training set, \( 5.8 \times 10^4 \) trigram models were obtained. As word pattern models, we extracted \( 4.4 \times 10^4 \) level 1 patterns and \( 5.3 \times 10^4 \) level 2 patterns from parse trees. In addition, we made a combination set of level 1 and 2 patterns. This pattern set had \( 8.7 \times 10^4 \) patterns.

The multi-class composite bigram models [1] were used as the baseline language models. Hereafter, they are referred to as “Baseline.” These models are classified word n-grams and use composite words. Their word accuracy performance is equal to or better than conventional word trigram models [8].

Furthermore, S. Zhang et al. [4] used the Maximum Entropy algorithm, but it was not used in this paper for comparison. Since the linkgram models were word-pair models, both function words and boundary markers were deleted from word patterns, and the obtained word-pairs were defined as linkgram models, \( 6.1 \times 10^7 \) pairs were obtained.
Table 1: Test set perplexities, perplexity reduction rates, word accuracy rates, and error reduction rates.

<table>
<thead>
<tr>
<th></th>
<th>Perplexity</th>
<th>Perplexity Reduction Rates [%]</th>
<th>Word Accuracy [%]</th>
<th>Error Reduction Rates [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>–</td>
<td>20.79</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Trigram with boundary markers</td>
<td>–</td>
<td>15.91</td>
<td>23.45</td>
<td>88.12</td>
</tr>
<tr>
<td>Pattern (Level 1)</td>
<td>0.10</td>
<td>14.88</td>
<td>28.45</td>
<td>88.22</td>
</tr>
<tr>
<td>Pattern (Level 2)</td>
<td>0.06</td>
<td>14.73</td>
<td>29.17</td>
<td>88.28</td>
</tr>
<tr>
<td>Pattern (Level 1&amp;2)</td>
<td>0.06</td>
<td>14.86</td>
<td>28.53</td>
<td>88.28</td>
</tr>
<tr>
<td>Linkgram</td>
<td>0.10</td>
<td>14.85</td>
<td>28.56</td>
<td>88.18</td>
</tr>
</tbody>
</table>

3.2. Experimental results

3.2.1. Perplexity

Figure 5 shows test set perplexities for various values of constant λ. The term “Baseline” denotes the baseline models, and “Trigram with boundary markers” means the trigram models with bigram markers. “Pattern (Level 1)” denotes the combination of the level 1 pattern models and trigram models with boundary markers. “Pattern (Level 2)” and “Pattern (Level 1&2)” mean the combination of level 2 pattern models and trigram models with boundary markers, and the combination of level 1, level 2 pattern models, and trigram models with boundary markers, respectively.

Although it is difficult to compare “Baseline” with the other models in the perplexity results, the trend of the perplexity performance is the same as that of the word accuracy performances. Therefore, it is reasonable to compare them in the perplexity results. The perplexity improvements from “Baseline” were 23.45% for “Trigram with boundary markers,” 28.45% for “Pattern (Level 1),” 34.02% for “Pattern (Level 2),” and 28.94% for “Pattern (Level 1&2).” “Pattern (Level 2)” obtained the best performance among these models. The long word patterns of “Pattern (Level 2)” are more effective than the other word patterns. The linkgram models were almost the same as “Pattern (Level 1)” and “Pattern (Level 1&2)” for optimal λ. These models were slightly better than “Pattern (Level 1)” and “Pattern (Level 1&2)” for larger pattern model weights. This is because linkgram models just have word-pairs, and they are robust, while pattern models have more than two word sets, which might corrupt the performance if the pattern weight is too large.

3.2.2. Word accuracy

Table 1 shows test set perplexities, perplexity reduction rates, word accuracy rates, and error reduction rates for optimal λ. The optimal λs were different from the λs for the best perplexities, especially for “Pattern (Level 2).” First, using “Trigram with boundary markers,” an 8.26% error reduction rate was obtained. This showed that it is useful to analyze and to include constituent boundary words into a language model. Second, the word pattern models were found to be slightly better than the trigram models with boundary markers. The combination of these models produced the best result in this experiment, which was a 9.50% error reduction rate. The word pattern models were also better than the linkgram models. Although the linkgram models could represent relations between long distance words, the word pattern models had stronger syntactic constraints.

The improvements by the word pattern models were not so large. This is because the evaluation sentences were too short at 9.6 words per sentence, while the training sentences had 13.9 words per sentence. Therefore, the word pattern models will be more effective for longer sentences.

Furthermore, the errors of some function words were reduced by these new language models. Since most function words are short, their acoustic reliabilities are sometimes not so high. However, the local syntactic constraints of the proposed language models are especially effective for function words.

4. Conclusions

We have proposed a new speech recognition scheme using three linguistic constraints. To extract phrase structures, the proposed language models, i.e., n-gram models with boundary words and word pattern models, are made by using the parser of TDMT. These new language models can represent phrasal structures. Furthermore, since the word pattern models have headword co-occurrences, they have neighboring phrasal constraints. Experimental results show that the new recognition scheme reduces word errors 9.50% from the conventional scheme by using word neighboring characteristics only.

5. References