Integrating Layer Concept Information into N-gram Modeling for Spoken Language Understanding

Nick J.C. Wang, Jia-Lin Shen, and Ching-Ho Tsai
Department of Man-Machine Interface, R&D Center, Delta Electronics Inc.,
Taipei, Taiwan, R.O.C.
nick.jc.wang@delta.com.tw

Abstract

The paper presents a novel approach, integrating layer concept information into the trigram language model, to improve the understanding accuracy for spoken dialogue systems. With this approach, both the recognition accuracy and out-of-grammar problem can be largely improved. The concept error rate is therefore reduced. In the experiment using a real-world air-ticket information spoken dialogue system for Mandarin Chinese, a relative concept error rate reduction of 30% is achieved.

1. Introduction

A spoken dialogue system consists of speech recognition, language understanding, and dialogue management, as well as the front-end acoustic feature analysis and probably speech synthesis for speech output. It is a system composed by many components. Consequently, the performance of a spoken dialogue system depends on all components, as well as its interface design [1]. Our group in Delta Electronics Inc. has been working on the research and development of Mandarin spoken-language technologies for years and cooperating with MIT Spoken Language Systems group.

In MIT, a telephone-based conversational system, Mercury, for real-time flight information inquiry and booking was developed using the Galaxy architecture [2]. We have been developing a Chinese version of Mercury system since the beginning of year 2003, based on MIT’s spoken language technology framework and software environment. Our system interacts with the user over the phone through a natural conversation and delivers flight schedules and pricing information. Its vocabulary includes over 200 major city names worldwide and 23 major airline names. It was designed as a way of mix-initiative interactions between man and machine; hence, natural speaking in Mandarin could be understood. However, in our experience, longer utterances are still not easy to be understood, which causes the major problem on using the system.

Some issues are believed to be important for popularizing spoken dialogue systems. First of all, the operation and maintenance are expected to be reliable and cost-effective. Secondly, the maturity and reusability should be enhanced in order to improve the development efficiency. Last but not the least, the accuracy and naturalness must meet with users’ expectation.

Conventionally speech recognition and language understanding are interfaced by n-best word sequences or word graphs [3]. A long sentence with speech recognition errors or out-of-grammar expressions would cause parsing failures. A partial parsing strategy may help with this, if only the errors occur beyond the target concept phrases. However, the partial parsing sacrifices completeness and depth of analysis [4][5]. Our proposed approach integrates a layer of concept phrase information into the N-gram model for speech recognition. Therefore, the speech recognizer not only can output a sequence of words, but also some additional information of concept phrase boundaries. For instance, the recognizer will output "I would like <route> to go to San Francisco<route>'", where "<route>" and "</route>" show the beginning and the end of a concept phrase. Our experiment demonstrates a large reduction of the parsing failures by using the boundary information. In addition, the language training corpus for N-gram modeling is constructed as a two-layer Stochastic Context-Free Grammar (SCFG) in our approach: a ‘sentence-form’ layer and a concept phrase layer. The N-gram model trained with the two-layer corpus is shown to be able to deal with data sparseness. The overall improvement by using the proposed method is therefore significant.

The following section will explain our proposed approaches. Section 3 shows our experimental setup and results. Conclusion is made in the end.

2. A concept phrase layer

There are various choices of the concept phrase layer. In our initial attempt, we experiment only one-layer of two target ‘concept categories’: <time> and <route>. Each target concept category consists of a group of semantic attributes describing a similar concept. For example, <time> category may consist of many semantic attributes, such as time=>year, time=>month, time=>day, time=>hour, time=>minute, time=>period-in-a-day, and time=>interval.

Sentence: May you please give me a ticket from Taipei to Seoul at six thirty on September 20?

Abstractive form: May you please give me a ticket <time> <route> at six thirty on September 20

Figure 1. Decomposing phrases out of a sentence

A sequence of words in an utterance describing semantic attributes of the same concept category is deemed as a phrase. For example, as illustrated in Figure 1, in an utterance like “May you please give me a ticket from Taipei to Seoul at six thirty on September 20?” the <time> phrase “at six thirty on September 20” and the <route> phrase “from Taipei to Seoul” could be identified. In another example “I want to fly
to Los Angeles on November 19 from Singapore”, the <route> phrases of “to fly to Los Angeles” and “from Singapore” are treated as two separate units because of their discontinuity. An ‘abstractive form’ is constructed by the remaining words in the sentence together with labels (like “<time>” and “<route>”) that indicate the locations of the concept phrases. Taking the above two sentences as examples, the abstractive forms of them are “May you please give me a ticket <route> <time>? and “I want <route> <time>”, respectively. In our data, most of the target phrases are easily identified.

The language training data is processed by the following way. Words in <time> and <route> phrases are labeled with <time> and <route> manually. For example, “from<route> Taipei<route>” and “September<time> 20<time>”. The attachment of labels slightly enlarges the vocabulary for the recognizer. Those concept phrases compose <time> and <route> phrase sets, separately. Abstractive forms compose a ‘sentence-form’ set. These sets will be used as a two-layer SCFG in the following section.

2.1. The two-layer-corpus trigram modeling

The multi-layer stochastic approach is popular in natural language understanding as in [6] and [7]. In the paper, we experiment the use of multi-layer stochastic approach to construct the corpus for N-gram modeling. In addition, the word-category based trigram modeling is adopted for the experiment because of its advantages on dealing with data sparseness [8].

Sentence-form set:
\[ S \rightarrow \text{I would like <route> <time>}. \]
\[ S \rightarrow \ldots \]

<Route> phrase set:
\[ <route> \rightarrow \text{to<route> go<route> to<route> Boston<route>}. \]
\[ <route> \rightarrow \text{to<route> fly<route> to<route> Paris<route>}. \]

<Route> phrase set:
\[ <time> \rightarrow \text{today<time> September<time> 20<time>}. \]
\[ <time> \rightarrow \text{tomorrow<time> morning<time>}. \]
\[ <time> \rightarrow \ldots \]

![Figure 2. The two-layer corpus](image)

I would like to <route> go<route> to<route> Boston<route> on<time> September<time> 20<time>.
I would like to <route> fly<route> to<route> Paris<route> on<time> September<time> 20<time>.
I would like to <route> go<route> to<route> Boston<route> tomorrow<time> morning<time>.
I would like to <route> fly<route> to<route> Paris<route> tomorrow<time> morning<time>.

![Figure 3. The derived sentences of the 2-layer corpus](image)

The corpus is constructed in the form of a two-layer SCFG, as illustrated in Figure 2. In the first layer, there are rules from the start symbol S leading to all abstractive forms. In the second layer, there are rules from the only two non-terminals “<route>” and “<time>” leading to all <route> and <time> phrases, respectively. A large number of grammatical sentences can be derived from the above rules, as illustrated in Figure 3. These sentences are then used to train the N-gram model. The resulted N-gram model inherits the property of two-layer statistics and embraces longer distance dependency.

The underlining assumption is that every sentence with a form like ‘I would like <route> <time>’ should share its observation with all grammatical sentences in the same form. Therefore, all grammatical sentences originated from the two-layer corpus should deal with the data sparseness problem.

Our N-gram model is estimated in conventional way. Hence, the resulted N-gram model has the same format as the conventional one and any conventional N-gram search engine can easily adopt it. An earlier paper has discussed a two-layer bigram model [9]. The unified language model of N-grams and Stochastic Finite State Automata was discussed in [10]. The unified language model of N-grams and SCFG was discussed in [11][12]. They all need a specialized recognizer.

The derivation of all grammatical sentences may generate a huge file of sentences. An alternative is to derive N-gram counts of all grammatical sentences. To begin with, the texts of sentence-form set and of concept phrase sets would be counted separately. Derivations of the non-terminal phrase labels in the sentence-form counts by their concept phrases are carried on element by element (from left to right). Besides, the derivation of any non-terminal X will depend on the length of the replacement word sequence from the phrase sets.

Firstly, as X is the first element in the triplet (X, u2, u3) of a count \( n_{d}(X, u2, u3) \), the derivation of the count will generate three possible branches, as shown in Equation 1 to 3: replacement of X by single words \( x_1 \) (i.e. length=1) if observing \( n_{d}(x_1, e) \), where \( e \) is the phrase ending mark; replacement by word pairs \( (x_1, x_2) \) (i.e. length=2) if observing \( n_{d}(x_1, x_2, e) \); and replacement by word triplets \( (x_1, x_2, x_3) \) (i.e. length=3) if observing \( n_{d}(x_1, x_2, x_3) \).

\[
\begin{align*}
\text{n}(x_1, u_2, u_3) &= n_{d}(X, u_2, u_3) \cdot n_{x}(x_1, e) / n_{x} \\
\text{n}(x_1, x_2, u_3) &= n_{d}(X, x_2, u_3) \cdot n_{x}(x_1, x_2, e) / n_{x} \\
\text{n}(x_1, x_2, x_3) &= n_{d}(X, x_2, x_3) \cdot n_{x}(x_1, x_2, x_3) / n_{x}
\end{align*}
\]

\( u_2 \) and \( u_3 \) denote either a word or a non-terminal node. \( n_{d}(.) \) and \( n_{d}(.) \) are the counts in sentence-form set and in X phrase set, respectively. \( n_{x} \) is the total number of X phrases.

Secondly, as X is the second element in the triplet \( u_1, X, u_3 \) of a count \( n_{d}(u_1, X, u_3) \), the derivation will generate two possible branches, as shown in Equation 4 to 5: replacement of X by single words \( x_2 \) (i.e. length=1) if observing \( n_{d}(b, x_2, e) \), where \( b \) is the phrase beginning mark; and replacement by word pairs \( (x_2, x_3) \) (i.e. length=2) if observing \( n_{d}(b, x_2, x_3) \).

\[
\begin{align*}
\text{n}(u_1, x_2, u_3) &= n_{d}(u_1, X, u_3) \cdot n_{x}(b, x_2, e) / n_{x} \\
\text{n}(u_1, x_2, x_3) &= n_{d}(u_1, X, u_3) \cdot n_{x}(b, x_2, x_3) / n_{x}
\end{align*}
\]

Finally, as X is the third element, the derivation generates replacement of X by single words \( x_3 \) (i.e. length=1) if observing \( n_{d}(b, x_3) \).

\[
\begin{align*}
\text{n}(u_1, x_2, x_3) &= n_{d}(u_1, u_2, X) \cdot n_{x}(b, x_2) / n_{x}
\end{align*}
\]
2.2. The two-pass parsing method

In the Mandarin Mercury system, a set of rules was written for complete full-sentence parsing. In our initial experiment on the proposed N-gram modeling, we use a newly designed two-pass parsing approach but with the existing rules.

Speech recognition generates a sequence of words and concept phrase boundaries like “May you please give me a ticket <route> from Taipei to Seoul <time> on September 20?” The complete parsing of the full sentences acts as the first pass. Once it fails, the second pass is to parse the joint phrase that is recognized with the help of boundary information, as illustrated in Figure 4. Taking the above sentence for instance, the recognized joint phrase “from Taipei to Seoul on September 20”, by combining <route> and <time> phrases, is parsed as the second pass, with the same parser and grammar. Different ways of using phrase boundary information will be experimented in the future.

**Figure 4.** The two-pass parsing, Parse A and B

Unavoidable recognition errors and incompleteness of grammar might result in understanding errors. Partial parsing approach is to decompose the parse into pieces of structure that can be reliably recovered with a small amount of syntactic information. It can recover syntactic information efficiently and reliably as the complete parsing fails. However, it sacrifices completeness and depth of analysis [4][5].

Our proposed approach provides a similar phrase-spotting ability as partial parsing does. Even so, an apparent difference exists between them: the former performs in speech recognition processing with entire acoustic and linguistic information, while the latter performs in language understanding processing with mainly linguistic information. It should be noticed that they could be applied either together or separately. In addition, better decisions may be made with more information.

3. Experiments

3.1. Experimental setup

Our system is composed of a segment-based speech recognizer SUMMIT [13] and a natural-language understanding system TINA [6] for spoken languages. The acoustic model for SUMMIT is trained using Mandarin telephony speech corpora MAT-2000 [14], while language modeling of N-grams for SUMMIT and of SCFG for TINA using 2,928 utterances collected through the Mandarin Mercury system. Vocabulary of the recognizer has 2,424 words in 186 categories.

Our language model training set was labeled with <route> and/or <time> phrases, while Set B is composed of neither. The average length of Set A is 5.59 words per sentence, which is 1.6 times of that of Set B. After decomposing target phrases, the average length of sentences decreases to 3.17. The average lengths of <time> and <route> phrases are 3.15 and 2.85, respectively. It shows that decomposition of the target phrases could largely shorten the sentences. Therefore, the parsing of decomposed sentences and phrases may outperform that of full sentences.

<table>
<thead>
<tr>
<th>Table 1. Statistics of the training utterances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set</td>
</tr>
<tr>
<td>Set A</td>
</tr>
<tr>
<td>Set B</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2. Statistics of Set A</th>
</tr>
</thead>
<tbody>
<tr>
<td>#Utterance</td>
</tr>
<tr>
<td>-----------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

A set of 3,389 utterances is used as the test set. The baseline system uses conventional category-based trigram modeling and complete parsing strategy (called ‘one-pass’ in the paper). Its performance is listed in Table 3: toneless syllable error rate (SER) 12.11%, concept error rate (CER) 21.03% and parsing failure rate (PFR) 10.33%.

<table>
<thead>
<tr>
<th>Table 3. The Baseline: 1-pass, no label</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>Baseline</td>
</tr>
</tbody>
</table>

3.2. Experiments using the conventional trigrams

A conventional trigram model is experimented in this subsection as a comparison to the two-layer-corpus trigram model, which is shown in the next subsection. Here are the steps for the experiment:

1. A conventional trigram model is trained with the labeled sentences for speech recognition.
2. A sequence of words and concept phrase boundary information are generated by speech recognition with the above trigram model.
3. The first pass parses the input of the whole word sequence.
4. Once if the first pass fails, the second pass parses the input of the joint phrase.

<table>
<thead>
<tr>
<th>Table 4. Experiments using conventional trigrams</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
</tr>
<tr>
<td>---</td>
</tr>
<tr>
<td>1-pass</td>
</tr>
<tr>
<td>2-pass</td>
</tr>
</tbody>
</table>

The SER in Table 4 is almost identical to that of the baseline in Table 3. A relative 0.25% difference is probably due to the little increase in vocabulary size by variants of label. After using the second pass parsing, PFR drops 53% relatively, from 10.39% to 4.90%, which contributes to a relative 10.4% CER reduction. The phrase boundaries from speech recognition seem useful and allow joint phrases to be parsed successfully.
3.3. Experiments using the two-layer-corpus trigrams
Here are the steps for the experiment:
1. A trigram model is trained with the two-layer corpus, according to the way discussed in subsection 2.1.
2. A sequence of words and concept phrase boundary information is generated by speech recognition with the above trigram model.
3. The first pass parses the input of the whole word sequence.
4. Once if the first pass fails, the second pass parses the input of the joint phrase.

Table 5. Experiments using two-layer-corpus trigrams

<table>
<thead>
<tr>
<th></th>
<th>SER</th>
<th>CER</th>
<th>PFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-pass</td>
<td>11.24</td>
<td>15.93</td>
<td>4.60</td>
</tr>
<tr>
<td>2-pass</td>
<td>11.24</td>
<td>14.74</td>
<td>2.42</td>
</tr>
</tbody>
</table>

After using the second pass parsing, it is observed of a significant 47% relative PFR reduction, similar to the result in subsection 3.2. Besides, the two-layer-corpus trigram model outperforms the conventional one. CER drops 29.9% relatively to 14.74% after SER drops 7.2% relatively to 11.24%, as comparing the 2-pass performance to the baseline performance in Table 3. The layer-corpus trigram modeling shows improvements both in speech recognition and in language understanding, due to the sharing of lower-layer probabilities to deal with data sparseness and the additional phrases boundary information for the second pass parsing.

The following two reasons might explain the significant improvement by using the proposed two-layer-corpus trigram modeling. Firstly, the trigram model might be well trained within the scope of the phrase layer by the constraint of a smaller vocabulary size and a shorter context range. In addition, the trained trigrams of word sequences across target phrase boundaries are partly influenced by the trigrams of the abstractive-sentence layer, which has longer distance dependency. This management could be an effective way of using the training sentences. Besides, the proposed trigram modeling is still mainly based on data-driven approach, so as to rely less on the experts for grammars. Collection of the phrase sets across applications may make it more mature and enhance the efficiency of system development.

4. Conclusions
In the paper, we experiment the proposed approach of integrating one-layer concept information into N-gram modeling for spoken language understanding. It outperforms conventional way by a significant concept error rate reduction of around 30% in our Mandarin Mercury system. Firstly, it provides a robust way to spoken language understanding by parsing phrases instead of full sentences. Like partial parsing, it outperforms the complete parsing and salvages lots of parsing failures. Secondly, it provides a computational efficiency for parsing long utterances, because that target phrases within a long utterance could be decomposed and parsed jointly or individually. Thirdly, it improves the recognition performance by smoothing away the data sparseness problem via the construction of two-layer corpus for N-gram modeling. Finally, by the reusable concept phrase components the effectiveness and efficiency of system development could be enhanced.

5. Acknowledgements
The authors would like to thank the MIT-SLS group for the long-term valuable support to Delta Electronics on the research and development of the spoken language technologies. We would like to thank all colleagues in our group for a lot of valuable discussions and cooperation. Finally, we would like to express our gratitude to Alan Lee for his endeavor to develop the Mandarin Mercury system.

6. References