Spoken Language Understanding Using Layered N-gram Modeling

Nick J.C. Wang

Graduate Institute of Communication Engineering, National Taiwan University &
Department of Man-Machine Interface, R&D Center, Delta-Electronics Inc.
Taipei, Taiwan, R.O.C.
nick.jc.wang@delta.com.tw

Abstract

This paper presents an approach which integrates layer concept information into the trigram language model in order to improve the understanding accuracy for spoken dialogue systems and to improve the portability of the language modeling materials among different narrow-domain applications. With this approach, both the recognition accuracy and out-of-grammar problem can be largely improved, and the concept error rate is therefore reduced. In the experiments, using real-world air-ticket information spoken dialogue system for Mandarin Chinese, the relative concept error rate reductions from 20% to 30% are observed among systems given different sizes of language model training data. Furthermore, the layered N-gram modeling approach provides an efficient way of using existing chunk-phrase corpora to build a new application, so as to improve the portability of the language modeling materials. Our experiment shows that the use of time chunk-phrases from a similar domain can achieve about 90% of the concept error-rate reduction compared to that of the in-domain collected training data. It shows an initial N-gram model might be established rapidly with the help of a library of chunk-phrase corpora before exhaustedly collecting and transcribing application-specific dialogue utterances.

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 on its interface design [1].

The ASR group in Delta Electronics Inc. has been working on the research and development of Mandarin spoken-language technologies for years. We also cooperate 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 more than 200 major worldwide city names 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.

Some issues are believed to be important for popularizing spoken dialogue systems. In the beginning of applying a spoken language understanding system to a new application, two of the most important works is to collect an appropriate size of a language-training corpus for a statistical language model used in the speech-recognition component and to write thoroughly a set of grammar rules for a parser used in the language-understanding component. Normally, besides the cost in writing and maintenance of a sufficient set of grammar rules for the specific application, it takes a long time and a lot of money in the collection to obtain a sufficient set of training sentences both from many different dialogue intents and from many different speakers. Without a robust language model for speech recognition, the recognition accuracy of the spoken words would be worse. There would be consequent parsing failures, a poor understanding accuracy, and a low dialogue success rate.

Our previous studies have proven that the proposed approach has improved the understanding accuracy via sharing chunk-phrase observations in modeling N-gram probabilities for speech recognition and via using recognized concept chunk-phrase boundaries intelligently for language understanding. The paper is going to extend the study on its ability in dealing with data sparseness and the portability of chunk-phrase corpora to similar application domains, especially in the building of initial systems.

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

2. A concept phrase layer

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 hence integrates a layer of concept phrase information into the N-gram model for speech recognition [6][7]. Therefore, the speech recognizer not only can output a sequence of words, but also some additional information of concept phrase boundaries. Our experiments have demonstrated a large reduction of the parsing failures by using the boundary information. In addition, the language model training corpus for N-gram modeling is constructed as a two-layer Stochastic Context-

Free Grammar (SCFG) in our approach: a 'sentence-pattern' layer and a concept phrase layer. The N-gram model trained with the two-layer corpus is shown to be robust to data sparseness and is able to use data from a similar application for initialization.

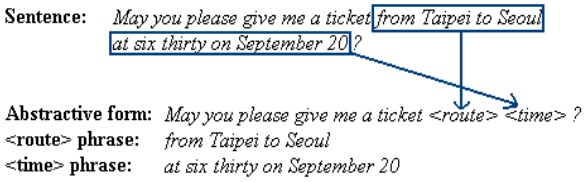

Figure 1. Decomposing phrases out of a sentence

There are various choices of the concept phrase layer. In our initial attempt, we experimented only one-layer of two target 'concept categories': <time> and <route>. 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. After extracting the phrases in the chunk layer, the abstractive form of the above sentence is simplified as *"May you please give me a ticket <route> <time>?"* In our statistical study of the training sentences [6][7], there are 68% among all containing either <time> or <route> chunks. The extraction of chunk phrases could simplify the whole sentence from 5.59 words in average to 3.17. The extracted chunk phrases have in average 3.15 words in the <time> phrase and 2.85 words in the <route> phrase. It is important to notice that current knowledge-based language understanding approaches heavily rely on the man-made grammar rules. Longer sentences usually lead to parsing failures due to incomplete grammar.

2.1. The two-layer-corpus trigram modeling

The multi-layer stochastic approach is popular in natural language understanding as in [8] and [9]. In this 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 because of its advantages on dealing with data sparseness [10].

Sentence-form set:
S → *I would like <route> <time>.*
S → ...

<route> phrase set:
<route> → *to<route> go<route> to<route> Boston<route>*
<route> → *to<route> fly<route> to<route> Paris<route>*
<route> → ...

<time> phrase set:
<time> → *on<time> September<time> 20<time>*
<time> → *tomorrow<time> morning<time>.*
<time> → ...

Figure 2. The two-layer corpus

The three corpora of <route> and <time> chunks and the sentence pattern work conceptually as a form of a two-layer SCFG, as illustrated in Figure 2. The chunk labels are saved for the space in the illustration. An attributed word like "to<route>" is treated differently from "to<time>".

In the first layer, there are rules from the start symbol *S* leading to all sentence patterns. In the second layer, there are rules from the two non-terminals "<route>" and "<time>" leading respectively to all <route> and <time> phrases. The expanding of the above two-layer rules can derive a large number of grammatical sentences, which are trained into the merged N-gram model by the following formulae. The resulting N-gram model not only inherits the property of two-layer statistics, but also embraces longer distance-dependency over the sentence-pattern layer.

Counts $n([PXQ]_N)$ in the merged N-gram count-tree are composed according to counts $n_S([PxQ]_N)$ in the sentence-pattern layer and counts $n_X([PxQ]_N)$ in the *'x'* chunk layer, where N_X is the total number of chunk *'x'* in the training data, X a sequence of words in chunk *'x'* to be expanded, token *b* the chunk (or sentence) beginning, token *e* the chunk (or sentence) ending, *P* a sequence of *n* words *'p_1,..., p_n'* or with *b* in the beginning *'b, p_1,..., p_{n-1}'*, *Q* another sequence of *m* words *'q_1,..., q_m'* or with *e* in the end *'q_1,..., q_{m-1}, e'*, and $[]_N$ a truncation operation to allow at most *N* words of statistics in N-gram modeling, as seen in Equation (1).

$$n([PXQ]_N) = n_S([PxQ]_N) \cdot n_X([bXe]_{N-N_P+1}) / N_X \quad (1)$$

Another type of counts $n([XQ]_N)$ in the merged N-gram count-tree, which begins with sub-phrase *X* in chunk, are composed according to Equation (2).

$$n([XQ]_N) = n_S([xQ]_N) \cdot n_X([Xe]_N) / N_X \quad (2)$$

The merging of the two-layer N-gram event-count trees generates a unified tree, which can be transformed to one N-gram probabilities model in the conventional format. Figure 3 illustrates the whole merging process. In our study, we found phrases longer than two words would perform better in its sharing of probabilities. Furthermore, a replacement of time chunk corpus from a similar-domain application could be experimented based on the proposed framework of layered N-gram modeling, which could hardly be possible in the traditional N-gram modeling approach.

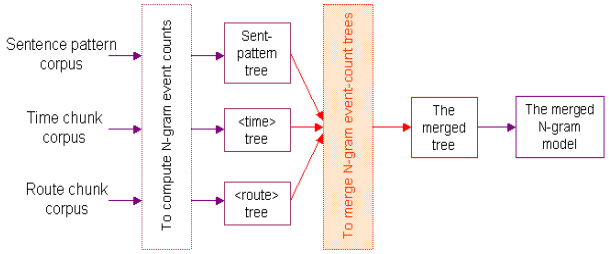

Figure 3. Two-layered N-gram modeling approach

Komatani explored similar combination of language models but limited on bigram format [11]. The underlining assumption is that every sentence with a form like *"I would like <route> <time>"* should share its observations with all grammatical sentences in the same form. Therefore, the merged N-gram model was trained conceptually based on more grammatical sentences of larger coverage and could be better in dealing with data sparseness. It is different from the phrase-based language modeling approach [12][13]. The latter extends the length of the context information in order to reduce the perplexity of the language model, while the former enhances the data sparseness over probability estimation.

Since the merged model is in the format of N-gram model, it could be easily adopted by many speech recognition systems without modification and preserves its robustness to spontaneous speech, especially as comparing to the Finite-State-Transducer decoder. On the contrary, earlier researches, such as the two-layer bigram model [14], the unified language model of N-grams and Stochastic Finite State Automata [15], and the unified language model of N-grams and SCFG [16][17], would need a specialized recognizer.

2.2. The two-pass parsing method

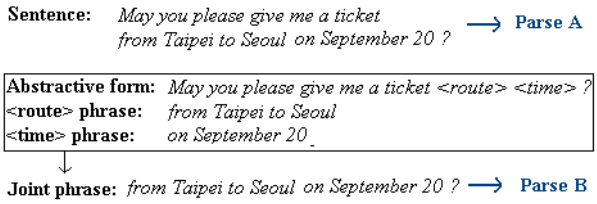

Figure 4. The two-pass parsing, Parse A and B

In the Mandarin Mercury system, a set of grammar rules were written to parse the whole sentence. In our initial experiment on the proposed N-gram modeling, we use a newly designed two-pass parsing approach but use the existing grammar 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</route> <time>on September 20</time>?"* 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.

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. 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 [18] and a natural-language understanding system TINA [8] for spoken languages. The acoustic model for SUMMIT is trained using Mandarin telephony speech corpora MAT-2000 [19], while language modeling of N-grams for SUMMIT and of SCFG for TINA using 2,928 utterances collected through the Mandarin Mercury system. The vocabulary of the recognizer includes 2,424 words in 186 categories.

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. Listed in Table 3 are the performances with different sizes of language model training data for comparison, where SER is toneless syllable error rate, CER concept error rate, and PFR parsing failure rate. With less training data the performance degrades rapidly.

Table 3. Conventional trigram modeling (baseline)

(%)	SER	PFR	CER
Baseline	11.92	10.56	20.82
1/2 data	12.44	11.77	21.75
1/4 data	13.62	15.11	25.43
1/8 data	14.51	17.17	27.15

3.2. Layered trigrams trained on different sizes of data

The subsection is going to study how the approach is dealing with data sparseness. Layered structure of the chunk and sentence-pattern corpora provides a flexible way of using all these tree corpora. The following experiments are going to try different sizes of chunk and/or sentence-pattern data in the N-gram modeling. Results are shown in Table 4.

Table 4. Layered trigrams given less <time> and <route> chunk phrases and/or less sentence patterns

(%)	SER	PFR	CER
L-trigram	11.26	2.98	14.67
1/2 all data	11.65	5.49	17.16
1/4 all data	12.14	7.61	18.82
1/8 all data	13.13	10.15	21.89
1/2 sent data	11.63	5.67	17.01
1/4 sent data	12.07	7.67	18.54
1/8 sent data	12.69	9.38	20.71
1/2 time data	11.42	3.10	15.04
1/4 time data	11.62	3.19	15.12
1/8 time data	12.42	3.72	16.49
1/2 route data	11.26	2.98	14.70
1/4 route data	11.22	2.95	14.64
1/8 route data	11.37	3.04	14.87

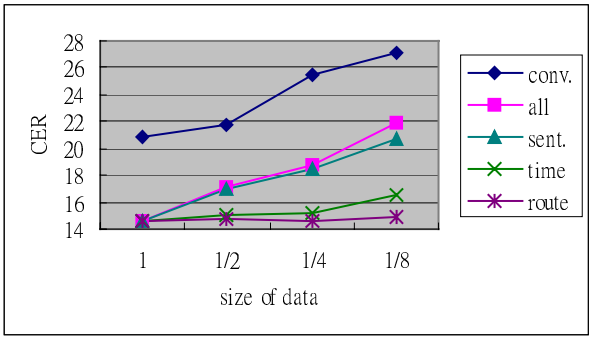

Figure 5. A comparison among different sizes of data

In Figure 5 we compare with conventional trigram modeling from Table 3. The layered trigram modeling approach outperforms the conventional trigram modeling with all different sizes of training data by the improvement on concept-error-rate reduction, by at least 20% CER reduction for the 'all' model. The 'time', 'route' and 'sent' results

demonstrate the dependency of the system performance over the subsets of the language model training data. The system seems to require a bigger size of the sentence-pattern corpus, as comparing to its requirement of 'time' and 'route' chunk phrases, probably because of its higher complexity.

3.3. Layered trigrams trained with chunk phrases from a similar application

One of our original targets about the proposed method is to overcome the difficulty of collecting language-training data. Like a divide-and-conquer approach, a sufficient set of language model training materials can be obtained by collecting sufficient subsets of chunk phrases and a sentence-pattern corpus.

In this subsection, we experiment a replacement of time chunk corpus to simulate an initial language model trained on materials from different applications. A set of time chunk phrases was extracted from a train information inquiry and booking system. Different time chunk event-count trees were merged with the other in-domain data, like route chunk event-count tree and sentence-pattern event-count tree, to train the final trigram models in our experiments. The layered N-gram model trained on the train-time chunk corpus is called 'train' model in the below table, while the model trained on hardly any time chunk phrase is called 'no' model. 'Flight' model is the one with in-domain time chunk corpus. Table 5 compares their results. As expected, the 'flight' one gives a good performance, SER 11.26% and CER 14.67% in our experiment, since its training set has no mismatch to its test set. Our main concern is how the 'train' one works. It gives us an impressive result with SER 12.59% and CER 16.68%. They are approaching to that of the 'flight' one as comparing the concept-error-rate reduction (CER-R), based on the 'no' one's SER 20.67% and CER 36.76% with no time chunk data. The experimental result shows the use of 'train' time chunk corpus reduces the concept-error-rate by 54.6% relatively, while the use of 'flight' time chunk corpus reduces 60.1%. The 'train' time corpus achieves relatively 90% of what the 'flight' one does. It demonstrates quite a good performance as a bootstrapping approach to building an initial system. The author has tried to merge both time-chunk corpora from two applications to train a layered trigram model. However, it gave no help in comparison to the model purely based on 'flight' time-chunk. More complicated framework like using language model adaptation could be studied in the future.

Table 5. Layered trigrams training on <time> chunk corpora from a different application

(%)	SER	PFR	CER	CER-R
No	20.67	8.88	36.76	-
Flight	11.26	2.98	14.67	60.1%
Train	12.59	3.19	16.68	54.6%

4. Conclusion

In the paper, we experiment the proposed layered N-gram modeling approach using time and route chunk information for spoken language understanding. It outperforms conventional way by significant concept error rate reductions, more than 20%, in our Mandarin Mercury system. It is also more robust to data sparseness. Besides, an effectiveness and efficiency way of system development is demonstrated by using portable chunk phrase corpora in our preliminary study.

5. Acknowledgement

The authors would like to thank the speech group in Delta Electronics for the development of Mandarin flight information dialogue system, and the MIT-SLS group for their valuable support on the system and the spoken language understanding technology.

6. References

[1] Zue, V. W. and Glass, J. R., "Conversational Interface: Advances and Challenges", *Proc. IEEE, Special Issue on Spoken Language Processing,* Vol. 88, August 2000.

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

[3] Giachin, E. and McGlashan, S., "Spoken Language Dialogue Systems", chapter three in *Corpus-Based Methods in Language and Speech Processing,* edited by Yong, S. and Bloothooft, G., published by Kluwer Academic, 1997.

[4] Abney, S., "Part-of-Speech Tagging and Partial Parsing", chapter four in *Corpus-Based Methods in Language and Speech Processing,* see above.

[5] Kellner, A., Rueber, B., Seide, F. and Tran, B.-H., "PADIS – an Automatic Telephone Switchboard and Directory Information System", *Speech Communication,* 1996.

[6] Wang, N. J.-C., Shen, J.-L., and Tsai, C.-H., "Integrating Layer Concept Information into N-gram Modeling for Spoken Language Understanding", *Proc. ICSLP,* 2004.

[7] Wang, N. J.-C., "Integrating Multiple Layers of Concept Information into N-gram Modeling for Spoken Language Understanding", *Proc. ICASSP,* 2005.

[8] Seneff, S., "TINA: A natural language system for spoken language applications," *Computational Linguistics*, vol. 18, no. 1., pp. 61-86, March 1992.

[9] Pla, F., Molina, A., Sanchis, E., Segarra, E. and Garcia, F., "Language Understanding Using Two-Level Stochastic Models with POS and Semantic Units", *Text Speech and Dialogue*, pp. 403-409, 2001.

[10] Brown, P. F., Pietra, V. J. D., deSouza, P. V., Lai, J. C. and Mercer, R. L., "Class-based n-gram Models of Natural Language", *Computational Linguistics* 18(4) pp. 467-479.

[11] Komatani, K., Tanaka, K., Kashima, H and Kawahara, T., "Domain-independent spoken dialogue platform using key-phrase spotting based on combined language model.", *Proc. Eurospeech,* 2001.

[12] Heeman, P. A. and Damnati, G., "Deriving Phrase-based Language Models", *Proc. ASRU,* 1997.

[13] Kuo, H.-K. J. and Reichl, W., "Phrase-based Language Models for Speech Recognition", *Proc. Eurospeech,* 1999.

[14] Goblirsch, D. M., "Viterbi Beam Search with Layered Bigrams", *Proc. ICSLP,* 1996.

[15] Nasar A, et al, "A Language Model Combining N-grams and stochastic Finite State Automata", *Proc. Eurospeech,* 1999.

[16] Wang, Y. Y., Mahajan, M. and Huang, X. "A Unified Context-Free Grammar and N-gram model for Spoken Language Processing", *Proc. ICASSP 2000.*

[17] Wang, K. "Semantics Synchronous Understanding for Robust Spoken Language Applications", *Proc. ASRU, 2003.*

[18] Glass, J., Hazen, T.J. and Hetherington, L., "Real-time telephone-based speech recognition in the JUPITER domain", *Proc. ICASSP,* Phoenix, AZ, March 1999.

[19] Wang, H. C., Seide, F., Tseng, C. Y. and Lee, L.-S., "MAT-2000 – Design, Collection, and Validation of a Mandarin 2000-Speaker Telephony Speech Database", *Proc. ICSLP,* Beijing, 2000.