# A UNIFIED CONTEXT-FREE GRAMMAR AND N-GRAM MODEL FOR SPOKEN LANGUAGE PROCESSING

*Ye-Yi Wang, Milind Mahajan, and Xuedong Huang**

Speech Technology Group
Microsoft Research
Redmond, Washington 98052, USA
http://research.microsoft.com/stg

## ABSTRACT

While context-free grammars (CFGs) remain as one of the most important formalisms for interpreting natural language, word n-gram models are surprisingly powerful for domain-independent applications. We propose to unify these two formalisms for both speech recognition and spoken language understanding (SLU). With portability as the major problem, we incorporated domain-specific CFGs into a domain-independent n-gram model that can improve generalizability of the CFG and specificity of the n-gram. In our experiments, the unified model can significantly reduce the test set perplexity from 378 to 90 in comparison with a domain-independent word trigram. The unified model converges well when the domain-specific data becomes available. The perplexity can be further reduced from 90 to 65 with a limited amount of domain-specific data. While we have demonstrated excellent portability, the full potential of our approach lies in its unified recognition and understanding that we are investigating.

## 1. INTRODUCTION

For the given speech signal **X**, spoken language understanding task is to find out the corresponding action $\mathbf{A}^*$, that satisfies the following equation:

$$\begin{aligned}\mathbf{A}^* &= \arg\ \max_{\mathbf{A}}\ P(\mathbf{A}\,|\,\mathbf{X}) \\ &\cong \arg\ \max_{\mathbf{A,S,W}}\ P(\mathbf{A}\,|\,\mathbf{S})\,P(\mathbf{S}\,|\,\mathbf{W})\,P(\mathbf{W}\,|\,\mathbf{X})\end{aligned} \quad (1)$$

where **A** stands for actions from dialog manager, **S** stands for semantic objects which are generated from a semantic parser [1, 2], and $\mathbf{W} = w_1 w_2 \ldots w_n$ is the word sequence from a speech recognizer [3]. Equation (1) indicates that we need to have a unified decoder from speech to understanding.

The goal of the language model (LM) $P(\mathbf{W})$ is to provide adequate information for predicting the likely word sequence. This can not only constrain the search space but also dramatically improve the accuracy of speech recognition. The CFG is not only powerful enough to describe most of the structure in spoken language but also restrictive enough to have efficient parsers. $P(\mathbf{W})$ can be regarded as 1 or 0 depending upon whether the word sequence is accepted or rejected by the grammar. While the CFG provides us with a deeper structure, it is still inappropriate for robust spoken language processing since the grammar is *almost always incomplete.* A CFG-based system is only good when you know what sentences to speak, which diminishes the value and usability of the system. The advantage of CFG's structured analysis is thus nullified by the poor coverage in most real applications. For application developers, it is also often highly labor-intensive to create CFGs.

On the other hand, grammaticality is irrelevant for the n-gram model. Because it can be trained with a large amount of data, the n-word dependency can often accommodate both syntactic and semantic shallow structure seamlessly. The prerequisite of this approach is that we must have a sufficient amount of training data. The problem for n-gram models is that we need a lot of data and the model may not be specific enough.

Nasr *et al.* [4] have considered a new unified language model that is composed of several local models and a general model linking the local models together. The local model used in their system is based on the stochastic FSA which is estimated from the training corpora. This approach still faces the portability problem, as it is hard to get domain-specific data to estimate these stochastic FSAs. Others [5-7] also considered a similar model using CFGs but once again, there is no clear way to leverage domain-independent LMs for domain-specific applications under the same probabilistic framework. In addition, none of these systems considered tightly integrating speech recognition (**X** to **W**), parsing (**W** to **S**), and dialog management (**S** to **A**) as illustrated in Equation (1).

## 2. A UNIFIED LANGUAGE MODEL

Our unified language model is trying to take advantage of both rule-based and data-driven approaches. We want to come up with the method that is the best in terms of not only performance but also portability. Let's consider the following training sentences:

*Meeting at three with Zhou Li.*
*Meeting at four PM with Derek.*

If we use a word trigram, we will estimate *P(Zhou|three with)* and *P(Derek|PM with)* etc. There is no way we can capture the needed long-span semantic information in the training data. A unified model will have a set of CFGs that can capture the semantic structure of the domain. For the example listed here, we may have CFGs for {name} and {time} respectively. We can then use our NL engine to parse the training data we used for training our trigram to spot all the potential semantic structures in the training data. The training sentences now look like:

*Meeting {at three:TIME} with {Zhou Li:NAME}*
*Meeting {at four PM:TIME} with {Derek: NAME}*

With analyzed training data, we can estimate our n-gram probabilities as usual. We will have probabilities such as P({name}|{time} with) instead of *P(Zhou|three with)*, which is more meaningful and accurate. Inside each CFG, we can also derive P("Zhou Li"|{name}) and P("four PM”|{time}) from the

* Alphabetically reversed order

0-7803-6293-4/00/$10.00 ©2000 IEEE.

existing n-gram (n-gram probability inheritance) so that they are normalized. If we add a new name to the existing {name} CFG, we can use the existing n-gram probabilities to renormalize our CFGs for the new name. The new approach can be regarded as a standard n-gram in which the vocabulary consists of words and structured classes. The structured class can be very simple such as {date}, {time}, and {name} or can be very complicated such as a CFG that contains deep structured information. Probability of a word or class will depend on the previous words or CFG classes.

Inside each CFG, we can use the standard probabilistic CFG. However, without real data to estimate these probabilities, there is no easy way to derive the probability for each production rule. In addition, the context-free nature of probabilistic CFGs may not offer any real advantage over n-gram models which have strong local context constraints. We therefore investigated how the CFGs can inherit probability from a (possibly general) word n-gram LM.

## 3. PROBABILITY INHERITANCE

Formally, an input utterance $\mathbf{W} = w_1 w_2 ... w_n$ can be segmented into a sequence $\mathbf{T} = t_1 t_2 ... t_m$ where each $t_i$ is either a word in **W** or a CFG non-terminal that covers a sequence of words $\bar{u}_{t_i}$ in **W**. The likelihood of **W** under the segmentation **T** is therefore

$$P(\mathbf{W},\mathbf{T}) = \prod_{i=1}^{m} P(t_i \mid t_{i-1}, t_{i-2}) \prod_{i=1}^{m} P(\bar{u}_{t_i} \mid t_i) \qquad (2)$$

In addition to trigram probabilities, we need to include $P(\bar{u}_{t_i} \mid t_i)$, the likelihood of generating a word sequence $\bar{u}_{t_i} = [u_{t_i 1} u_{t_i 2} ... u_{t_i k}]$ from the CFG non-terminal $t_i$. In the case when $t_i$ itself is a word ($\bar{u}_{t_i} = [t_i]$), $P(\bar{u}_{t_i} \mid t_i) = 1$. Otherwise, $P(\bar{u}_{t_i} \mid t_i)$ can be obtained by predicating each word in the sequence on its word history:

$$P(\bar{u}_{t_i} \mid t_i) = \left[ \prod_{l=1}^{\|\bar{u}_{t_i}\|} P(u_{t_i l} \mid u_{t_i 1}, ..., u_{t_i l-1}) \right] P(</s> \mid \bar{u}_{t_i}) \qquad (3)$$

Here </s> represents the special end-of-sentence word. Three different methods are used to calculate the likelihood of a word given history inside a CFG non-terminal.

### 3.1 Uniform Distribution

A history $h = u_{t_i 1} u_{t_i 2} ... u_{t_i l-1}$ corresponds to a set $Q(h)$, where each element in the set is a CFG state generating the initial $l-1$ words in the history from the non-terminal $t_i$. A CFG state constrains the possible words that can follow the history. The union of the word sets for all of the CFG states in $Q(h)$, $W_Q(h)$ defines all legal words (including the symbol "</s>" for exiting the non-terminal $t_i$ if $t_i \overset{*}{\Rightarrow} u_{t_i 1} u_{t_i 2} ... u_{t_i l-1}$) that can follow the history according to the CFG constraints. The likelihood of observing $u_{t_i l}$ following the history can be estimated by:

$$P(u_{t_i l} \mid h) = 1 / \|W_Q(h)\| . \qquad (4)$$

### 3.2 Inherited Word N-grams

The uniform model does not capture the empirical word distribution underneath a CFG non-terminal. A better alternative is to inherit existing domain-independent n-gram probabilities. These probabilities need to be appropriately normalized in the same probability space. Thus we have:

$$P(u_{t_i l} \mid h) = \frac{P(u_{t_i l} \mid u_{t_i l-2}, u_{t_i l-1})}{\sum_{w \in W_Q(h)} P(w \mid u_{t_i l-2}, u_{t_i l-1})} \qquad (5)$$

### 3.3 CFG-Specific Inheritance

Another way to improve the modeling of word sequence covered by a specific CFG non-terminal is to use a specific n-gram LM $P_t(w_n \mid w_{n-2}, w_{n-1})$ for each non-terminal *t*. The normalization is performed in the same way as in Equation (5).

Multiple segmentations may be available for **W** due to the ambiguity of natural language. The likelihood of **W** is therefore the sum over all segmentations S(**W**):

$$P(\mathbf{W}) = \sum_{\mathbf{T} \in S(\mathbf{W})} P(\mathbf{W},\mathbf{T}) \qquad (6)$$

## 4. A UNIFIED DECODER

It is desirable to extend this framework further to unify both CSR and SLU instead of the current two-pass SLU systems. As illustrated in Equation (1), the full potential of our new approach is that we can unify a number of components (speech recognizer, parser, and dialog manager) under the same probabilistic framework for optimal performance, which integrates the traditional rule-based NL approach and the most powerful data-based NL model (n-gram) seamlessly for both speech recognition and understanding.

If we can identify these CFGs in the decoder, the need for a separate NL parser and speech recognizer may diminish. The advantage of our unified approach is that we can spot semantic concepts directly from the speech signal.

Our current Whisper decoder [3] can only support either CFGs or word n-grams. These two grammars are mutually exclusive. We are in the process of changing the decoder so that we can embed CFGs in the n-gram search framework to take advantage of our unified language model.

## 5. MIPAD

We are developing a multimodal interactive pad (MiPad) that offers a conversational, multi-modal interface to Personal Information Manager (PIM) functionality, including calendar, task list and e-mail. The ultimate goal is an interaction model that spans across a number of different platforms and users. The initial target device is in the palmtop form factor, and is intended for use by mobile professionals. We have chosen this as the platform because it is clearly a useful tool and has several opportunities for improvements. With existing palmtop PDAs, it is very difficult to enter large amount of text, to fill a form, and to issue commands that contain multiple parameters. Multimodal interaction with speech and pen can help address these problems, which can significantly improve the usability with the *Tap and Talk* interface.

MiPad uses the Whisper speech recognizer [3] with a 60,000 word vocabulary. The system can be adapted to the user for improved performance. The current understanding system is based on our robust parser [8] and event-driven dialog manager [9].

# 6. EXPERIMENTAL RESULTS

In the preliminary study, we only focused our experiments on portability of our domain-independent language model. We investigated how to build a domain specific language model without using domain specific data.

## 6.1 Baseline System

We built a general purpose trigram LM with vocabulary of 2,000 words. The model was trained with the same data as that used for the Microsoft Dictation trigram. We used text corpora from newspapers, TV program transcripts, and memos. The training data has more than 2 billion words. The test set consists of 2,000 sentences related to MiPad's PIM applications such as scheduling a meeting, finding information from the contact list, and email. We collected these sentences in-house for the development of MiPad. Some of the training sets which we used in our experiments are so small that some words in the vocabulary never occur in the training set. Therefore, we always interpolated all LMs with a uniform word distribution with a small interpolation coefficient (0.05) to provide the necessary smoothing.

The perplexity of our baseline Microsoft dictation language model on our MiPad test data is 378[1]. There is a clear mismatch between the dictation language model and conversational MiPad test data.

## 6.2 A TFIDF Model

The general purpose trigram was trained with a large variety of data. A majority of the data is likely to be irrelevant to our domain. Thus a topic-dependent language model such as that suggested in [10] can be used to select relevant text materials to build a more domain-specific LM.

We used an Information Retrieval (IR) technique to extract more relevant data from the training set. We ran the CFG in the generative mode to generate "sentences" and used them as a query for IR [10]. For each sentence in the training data, its similarity to the query is calculated using the cosine similarity measure of the respective TFIDF vectors. Only those sentences that are similar to the query were used for training the trigram. The perplexity of the trigram trained on the filtered data (henceforth TFIDF model) is 271.

## 6.3 A CFG-Derived Word Trigram

We cannot use CFGs directly to evaluate the perplexity since a large number of sentences are not covered by our CFGs. Instead, we used our CFGs to generate sentences and used these sentences to estimate a word trigram.

The perplexity of the CFG-derived trigram LM is 207, which indicates that the coverage of the CFG alone is indeed limited.

## 6.4 An Interpolated Trigram

We interpolated the TFIDF trigram LM and the CFG word trigram LM. Since we did not assume any domain specific data, 0.5 was used as the interpolation weight for the component LMs. The perplexity of the resulting LM is reduced to 112, which is a significant perplexity reduction over both the component LMs. Clearly the TFIDF data and the CFG-derived data contain highly complementary information.

## 6.5 A Unified Model

We parsed the data obtained from the aforementioned IR technique with our robust chart parser. A word/non-terminal trigram LM was trained with the parsed data. Since the occurrence of domain specific CFG non-terminals (like {date}, {time}, {appointments}) is much lower in the general data than in the domain specific data, we used the CFG again in the generative mode to obtain domain specific synthetic data that contained words and non-terminals. Starting from the top level CFG non-terminal, the procedure randomly decided whether to keep the non-terminal in the synthetic sentence or to expand it to sub-symbols according to the CFG rules for that non-terminal. A trigram LM was constructed on the synthetic data and it was interpolated with the unified model trained on the parsed TFIDF data. The interpolation weight for each component LM is 0.5.

Table 1 Comparison of language models on the MiPad test data when no domain-specific data is available

| *Language Model* | *Perplexity* |
|---|---|
| Baseline Trigram | 378 |
| TFIDF Trigram | 271 |
| CFG-derived Trigram | 207 |
| Interpolated Trigram | 112 |
| **Unified Language Model** | **90** |

Since many in-domain words are subsumed by CFG non-terminals, their probability of being a standalone word is underestimated. This is not very harmful if the CFG has good coverage. However, as we stated in the very beginning, high CFG coverage is not realistic for spoken language. To compensate for it, we interpolated the word/non-terminal LM described above with the word ngram model described in Section 6.4.

We investigated different methods of assigning the likelihood to a word sequence inside a CFG non-terminal, as discussed in Section 3. The best perplexity is 90, which is obtained from using inherited trigrams inside the CFG. The inherited trigram is CFG non-terminal specific as described in Section 3.3. The perplexity results are shown in Table 1.

[1] Using the standard DARPA NAB word trigram LM which has a larger vocabulary, the perplexity on this MiPad test set is more than 1000 while the typical Wall Street Journal text perplexity is about 100. This strongly indicates that there is a mismatch between these two domains.

### 6.6 Comparison with Domain Specific Models

We can train a domain specific trigram that should have much better performance in comparison with the domain-independent trigram. The key problem is we need to collect a large amount of training data, which is impractical for most application developers. For MiPad, we have collected 3,000 sentences and reserved 2,000 for testing. We used the other 1,000 utterances for training. The perplexity of the model is a reference for comparison with the model obtained without domain specific data.

Table 2 Comparison of language models on the MiPad test data when domain-specific data becomes available

| *Language Model* | *Perplexity* |
|---|---|
| Word Trigram | 186 |
| Interpolated Trigram | 91 |
| **Unified Language Model** | **65** |

Given the small amount of training data, we believed that the LM was likely to be under-trained. To improve the robustness, we interpolated the domain-specific trigram LM with CFG-derived trigram LM (Section 6.3). As shown in Table 2, the perplexity is significantly reduced to 65 with the unified model when limited amount of training data becomes available. In contrast, the interpolated word trigram has a much higher perplexity. This illustrates that our unified model can truly make more effective use of CFGs and domain-specific data than interpolated word trigram models.

## 7. DISCUSSION AND SUMMARY

Since we can have CFGs inherit n-gram probabilities, we can fully unify both CFGs and n-grams in the same probabilistic framework. When training data becomes available, the unified model is adaptable and it will converge to the best domain-specific structured n-gram language model. We can either adapt the system using new rules or data. When we port our system to a new domain, we can create some CFGs that may have limited coverage (as always), but the system can broaden the coverage of our CFGs automatically based on the n-gram language model. We can thus relatively easily port our SLU applications from one domain to another.

The full potential of the proposed approach lies in its unified recognition and understanding. As indicated in Equation (1), we believe that early use of semantic knowledge is very important to improve the robustness of the SLU system. We are in the process of systematically evaluating both the recognition and understanding performance in comparison to the conventional detached systems (speech recognition first and then SLU), which requires rewriting both the speech recognizer and SLU engine.

In our current approach, we have not used any deep linguistic concepts and our CFGs can be written and used by application developers who have domain-specific knowledge. This is important, as most application developers do not have any linguistic expertise. Furthermore, our architecture also provides a new framework to incorporate linguistics-driven NLP ideas in the future.

Our preliminary experiments indicate that the unified model could significantly improve the SLU system's portability, which has been a major problem for widespread application of spoken language technologies. The unified language model reduced the test set perplexity from 378 to 90 in comparison with a domain-independent word trigram. Two key components are responsible for such a dramatic perplexity reduction. The first one is the use of domain specific knowledge in CFGs. By interpolating a trigram derived from such CFGs with the domain-independent trigram, we can reduce the perplexity from 378 to 112, while the CFG-derived trigram alone has a much higher perplexity of 207. The second one is unification of CFGs and n-gram models, which further reduced the perplexity from 112 to 90.

When a limited amount of domain-specific data becomes available, the unified model offers further improved performance. The perplexity for the domain-specific word trigram was reduced from 186 to 91 when interpolated with the CFG-derived trigram. With the unified model, the perplexity was further reduced from 91 to 65.

## 8. REFERENCES

[1] Ward, W., *Understanding spontaneous speech: the Phoenix system.* Proceedings ICASSP. 1991: p. 365-367.

[2] Seneff, S. *The Use of Linguistic Hierarchies in Speech Understanding.* in *ICSLP.* 1998. Sydney, Australia.

[3] Huang, X., *et al., From Sphinx II to Whisper: Making Speech Recognition Usable*, in *Automatic Speech and Speaker Recognition*, C.H. Lee, F.K. Soong, and K.K. Paliwal, Editors. 1996, Klewer Academic Publishers: Norwell, MA. p. 481-508.

[4] Nasr, A., *et al. A Language Model Combining N-grams and Stochastic Finite State Automata.* in *Eurospeech.* 1999.

[5] Gillett, J. and W. Ward. *A Language Model Combining Trigrams and Stochastic Context-Free Grammars.* in *ICSLP.* 1998. Sydney, Australia.

[6] Moore, R., *et al, Combining Linguistic and Statistical Knowledge Sources in Natural-Language Processing for ATIS*, in *Proceedings of the ARPA Spoken Language Systems Technology Workshop.* 1995, Morgan Kaufmann, Los Altos, CA: Austin, Texas.

[7] Galescu, L., E.K. Ringger, and J. Allen. *Rapid Language Model Development for New Task Domains.* in *Proceedings of the ELRA First International Conference on Language Resources and Evaluation (LREC).* 1998. Granada, Spain.

[8] Wang, Y.-Y. *A Robust Parser For Spoken Language Understanding.* in *Eurospeech.* 1999. Hungary.

[9] Wang, K. *An Event Driven model for Dialogue Systems.* in *ICSLP.* 1998. Sydney, Australia.

[10] Mahajan, M., D. Beeferman, and X.D. Huang. *Improved Topic-Dependent Language Modeling Using Information Retrieval Techniques.* in *ICASSP.* 1999. Phoenix, AZ, USA.