A Word Graph Interface for a Flexible Concept Based Speech Understanding Framework

Kadri Hacioglu and Wayne Ward

Center for Spoken Language Research
University of Colorado at Boulder
{hacioglu,whw}@cslr.colorado.edu

Abstract

In this paper, we introduce a word graph interface between speech and natural language processing systems within a flexible speech understanding framework based on stochastic concept modeling augmented with background "filler" models. Each concept represents a set of phrases (written as a context free grammar (CFG)) with the same meaning, and is compiled into a stochastic recursive transition network (SRTN). The arcs (or rules) are tagged with probabilities after training. The filler models are used for phrases that are not covered by the concept networks. The structure in concept+filler sequences is captured by n-grams. The interface is implemented within the context of CU Communicator spoken dialog system. We investigate the effect of several different filler models and interpolation of complementary language models on the system performance. We report notable performance improvements compared to the baseline system. The gain in performance along with the efficiency and flexibility of the method motivates future work on the implementation of a tighter interface.

1. Introduction

The ultimate goal in a spoken dialog system is to understand what has been spoken and take the corresponding action. This suggests a system that maps input speech to actions. Except for very small size tasks, the present status of technology does not offer an effective and efficient solution to that problem. Therefore, we decompose the problem into two parts as speech understanding and action generation, and focus on the speech understanding part. The latter is a system which maps input speech to meaning represantation. This process is nowadays further decomposed into a speech recognizer (that provides a text transcription of the input speech) and a language understanding unit (that extracts meaning from the text) in state-of-the-art spoken dialog systems. Understanding speech is quite different from understanding text. Syntactic and semantic knowledge of language is required to be incorporated simultaneously whilst the input speech is processed. The ungrammatical constructs, filled pauses, repairs and ellipsis in a spoken language further complicates the task. In this research, we deal with this problem being motivated by

- dramatic increases in computer speed and memory
- efficient and effective implementations of recognition and parsing search algorithms [1, 2, 3].
- promising results from our semantically driven language modeling [4, 5]

The work is supported by DARPA through SPAWAR under grant #N66001-002-8906.

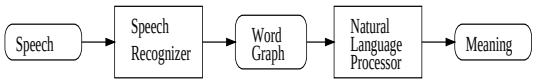

Figure 1: Extraction of meaning from speech using a word graph

To pave the way to our ultimate goal (direct mapping of speech to meaning) we first consider the extraction of meaning from input speech at word graph level. The generic diagram of the system is illustrated in Figure 1.

It can be considered as an extension of the speech understanding part of our CU Communicator dialog system [6] which extracts meaning from the speech recognizer's (CMU Sphinx-II's [1]) best hypothesis using a robust, heuristic parser called Phoenix [7]. In the new system we use a stochastic parser to convert a word graph into a concept graph. The concept graph is then searched by the Viterbi algorithm for the best concept sequence using a dialog context dependent concept language model along with the acoustic and rule probabilities computed in the previous passes. The detected concept sequence and the respective word sequence are passed to the Phoenix parser to extract meaning. The concept sequence constrains the semantic grammars used to parse the word sequence.

A similar work based on the concept modeling and a word graph interface has been reported in [8]. In that system, a concept bigram LM with a "garbage" filler model was used. In this paper, however, we investigate a dialog context dependent concept trigram LM with different filler models. Furthermore, we report experimental results obtained by interpolating the concept LM with a word or a class/word LM. We show that more precise filler models perform better than the "garbage" filler model and the interpolation improves the performance further. The interpolation of concept and word based LMs has been also studied in [9]. Our results show that the performance of the interpolation with class/word LM is slightly better than that of the interpolation with a word LM.

The paper is organized as follows. Section 2 introduces the mathematical framework. In section 3, we briefly explain the models used. The word graph interface is explained in Section 4. Experimental results are reported in Section 5. In Section 6, we discuss a possible implementation of a tightly coupled system as a future work.

2. Mathematical framework

We assume speech as a sequence of acoustic observations

$$A = a_1\ a_2\ \cdots a_T$$

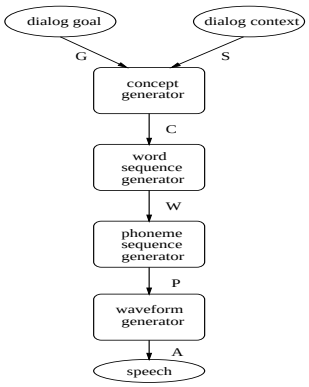

Figure 2: A speech production model

and meaning as a sequence of semantic units

$$M = m_1\ m_2\ \cdots m_L$$

A semantic unit is a (concept,value) pair, e.g. ([city_name], Denver). A value may be a word, a string of words or another concept. The latter allows arbitrary recursion accounting for the nested structure of a language. There are several ways of expressing the same concept in a sentence. That is, a concept is associated with a set of phrases where each phrase includes the value that can be tied to it or its subconcepts.

Our framework is based on the speech production model shown in Figure 2. It is a slightly modified version of the model described in [10]. The user is assumed to have a specific goal that does not change throughout the dialog. According to the goal and the dialog context the user first picks a set of concepts with respective values and then uses phrase generators associated with concepts to generate the word sequence. The word sequence is next mapped into a sequence of phones and converted into a speech signal by the user's vocal apparatus which we finally observe as a sequence of acoustic feature vectors.

The ultimate goal is to minimize the semantic error rate (SER). This can be accomplished (not exactly, though) by maximum a posteriori optimization:

$$M^* = \underset{M}{\text{argmax}}\ P(M/A,S) \tag{1}$$

where S denotes the dialog context. The modeling of meaning conditioned on acoustic observations is difficult if not impossible. Therefore, we introduce two other levels of knowledge, as sequence of words and phones, into the MAP optimization:

$$M^* = \underset{M}{\text{argmax}} \sum_{Ph} \sum_{W} P(M,W,Ph/A,S) \tag{2}$$

Assuming (i) Viterbi approximation, (ii) A is dependent only on Ph, (iii) Ph is dependent only on W and (iv) W is dependent only on M the final expression for the MAP optimization is

$$M^* = \underset{M}{\text{argmax}}\ \max_{W,Ph} P(A/Ph)P(Ph/W)P(W/M)P(M/S) \tag{3}$$

In (3) we identify four models:

- Semantic model: $P(M/S)$
- Syntactic model : $P(W/M)$
- Pronunciation (or lexical) model: $P(Ph/W)$
- Acoustic-phonetic model: $P(A/Ph)$

The semantic model is the a priori probabilities of semantic sequences conditioned on the dialog context. The syntactic model is the probability of word strings used to express a given semantic unit. The pronunciation model gives the probabilities of possible phonetic realizations of a word. The acoustic model is the probability for the occurance of acoustic feature observations given phones.

In a typical, moderate size task, like Air/Hotel/Car reservation, although the number of concepts is very small the number of semantic units could be very large due to the relatively large set of values. So, data sparsity is an issue in the modeling of the semantic model. However, during the MAP optimization in (3) the word sequence W is available as a by product. Therefore, to avoid the data sparsity problem to a certain extent, we focus only on concepts in the MAP optimization and get values from W by a "focused" parsing in a subsequent stage. This approach also avoids the use of complex meaning representations (tree structures for nested constructs) in the statistical models. We accordingly modify the MAP optimization in (3):

$$C^*, W^* = \underset{C,W}{\text{argmax}}\ \max_{Ph} P(A/Ph)P(Ph/W)P(W/C)P(C/S) \tag{4}$$

where $P(C/S)$ is the dialog context conditioned concept model. The focused parser , which is deterministic, extracts the meaning from W^* using the grammars constrained by C^*.

The preceding analysis assumes that the grammar written for the concepts covers the whole spoken sentence. A grammar with full coverage is hardly possible in practice, particularly for spoken language. For this reason, we augment the set of domain specific concepts with "filler" models to account for word patterns that are not covered by the grammar.

3. Description of models

In (4) one can distinguish between two modules. The first one, which computes $P(A/Ph)P(Ph/W)$, is the speech processing module and the second one, which computes $P(W/C)P(C/S)$, is the language processing module. In this section we explain the acoustic-phonetic, lexical and language models.

We use context dependent phone HMMs for $P(A/Ph)$. The HMMs are semi-continuous and distributions are shared among similar states. Lexical modeling is done by allowing multiple pronunciations of words in the lexicon. So, in our sytem we do not have an explicit pronunciation model. That is, $P(Ph/W)$=1 .

The concepts are classes of phrases with the same meaning. Put differently, a concept class is a set of all phrases that may be used to express that concept (e.g. [i_want], [arrive_loc]). Those classes are augmented with a "filler" model. Any input which is not covered by the concepts will be modeled by the "filler" model. We consider the following "filler" models:

1. a large set of single word concepts (Degenerate filler model)
2. a small set of single word concepts and a fairly small number of broad and unambigious part of speech (POS) classes. (D_POS filler model)
3. a single "garbage" concept. (Garbage model)

The examples in Figure 3 clearly illustrate the use of the models described above for a text input.

<s> I WANT TO FLY FROM MIAMI FLORIDA TO SYDNEY AUSTRALIA ON OCTOBER FIFTH </s>
<s> [i_want] [depart_loc] [arrive_loc] [date] </s>

<s> I DON'T TO FLY FROM MIAMI FLORIDA TO SYDNEY AFTER AREA ON OCTOBER FIFTH </s>
<s> [i] [don't] [depart_loc] [arrive_loc] [after] [area] [date] </s>
<s> [Pronoun] [Contraction] [depart_loc] [arrive_loc] [after] [Noun] [date] </s>
<s> [garbage] [depart_loc] [arrive_loc] [garbage] [date] </s>

Figure 3: Examples of parsing into concepts and filler models

The structure of the concept sequences is captured by an n-gram LM. Furthermore, the concept sequences are conditioned on the dialog context. Although there are several ways to define a dialog context, we select the last question prompted as the dialog context. It is simple and yet strongly predictive and constraining. Consequently , we need to train a seperate concept language model for each dialog context. Given the context S and $C = c_0 c_1 \cdots c_K, c_{K+1}$, the concept sequence probabilities are calculated as (for $n = 3$)

$$P(C/S) = P(c_1/ < s >, S) P(c_2/ < s >, c_1, S) \prod_{k=3}^{K+1} P(c_k/c_{k-2}, c_{k-1}, S)$$

where c_0 and c_{K+1} are for the sentence-begin and sentence-end symbols, respectively.

Each concept (except "garbage" and "degenerate" concepts) is written as a CFG and compiled into a stochastic recursive transition network (SRTN). The production rules are complete paths beginning from the start-node through the end-node in these nets. The probability of a complete path traversed through one or more SRTNs initiated by the top-level SRTN associated with the concept is the probability of the phrase that belongs to that concept. This probability is calculated as the multiplication of all arc probabilities that defines the path. That is,

$$\begin{aligned} P(W/C) &= \prod_{i=1}^{K} P(s_i/c_i) \\ &= \prod_{i=1}^{K} \prod_{j=1}^{M_i} P(r_j/c_i) \end{aligned}$$

where s_i is a substring in $W = w_1, w_2..w_L = s_1, ..s_2, s_K$ ($K \leq L$) and $r_1, r_2, ...r_{M_i}$ are the production rules that construct s_i. The concept and rule sequences are assumed to be unique in the above equations which is true for unambigious associations or Viterbi approximation. SCFG and n-gram probabilities are learned from a text corpus (parsed using heuristics) by simple counting and smoothing. Our semantic grammars have a low degree of ambiguity and therefore do not require computationally intensive stochastic training and parsing techniques. The "garbage phrase" probabilities are modeled by a "garbage concept" conditioned word bigram LM.

4. Word graph interface

The word graph is a directed acyclic graph (DAG). Nodes are uniquely labeled with frame numbers and word hypotheses that start at those frames. The arcs are labeled with the acoustic scores obtained during the first pass. The first pass is a frame synchronous tree lexicon Viterbi beam search. The language

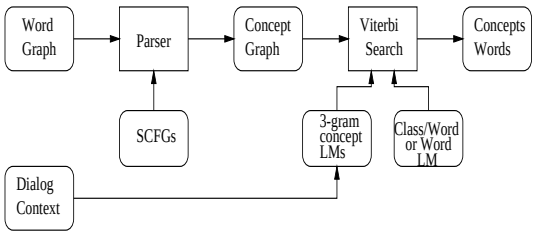

Figure 4: The system diagram

processing is based on a class/word trigram LM with bigram Viterbi recombination [1]. Within the framework of Section 2, the word graph interface can be considered as the following optimization:

$$C^*, W^* = \underset{C, W \in G_w}{\text{argmax}} P_A(W) P(W/C) P(C/S) \qquad (5)$$

where $P_A(W)$ is the acoustic score of the word sequence computed in the first pass and G_w denotes the word graph. We implement (5) as a two-stage process. In the first stage we convert the word graph into a concept graph where the possible gaps are filled with the filler models. In the second stage we search the concept graph for the best path. The concept graph is structured as a DAG with nodes containing concept pairs (this allows the optimal path search with a concept trigram LM) and respective phrases. The arc probabilities are the product of the acoustic probabilites associated with the word sequences covered by the concept pairs. The phrase probabilities from SCFGs are tagged to the nodes. The concept graph search can be considered as the following optimization within the model constraints:

$$C^* = \underset{C \in G_c}{\text{argmax}} P_T(C, W_C) P(C/S), \quad W^* = W_{C^*} \qquad (6)$$

where $P_T(C, W_C) = P_A(W_C) P(W_C/C)$ is the product of acoustic and phrase probabilites associated with the word sequence W_C uniquely defined by the concept sequence C and G_c denotes the concept graph.

Assuming that a word or a class/word based LM complements the language model provided by our language processing unit, we extended the interface to allow interpolation in log-linear domain at concept/phrase level. We selected the log-linear interpolation because of its better performance compared to the linear interpolation as reported in [5]. The block diagram of the interface is illustrated in Figure 4.

5. Experimental results

The models were developed and tested in the context of the CU Communicator dialog system which is used for flight, hotel and rental car reservations [6]. The text corpus was divided into two parts as training and test sets with 15220 and 1264 sentences, respectively. The test set is from a total of 72 calls made by the users selected by the National Institute of Standards (NIST). Of these, 44 callers were female and 28 were male. The test set was further divided into two parts. Each part, in turn, was used to optimize language and interpolation weights to be used for the other part. The results were reported as the average of the two results. The average sentence length of the corpus was 4 words (end-of-sentence was treated as a word). We identified 20 dialog contexts and labeled each sentence with the associated dialog context. During the experiments gender dependent (GD) acoustic models were used.

Table 1: Word error rate results of different filler models

Filler Model	WER
Degenerate	22.2%
D_POS	22.0%
Garbage	22.6%

We trained a dialog independent (DI) class based LM, a DI word based LM and several dialog dependent (DD) grammar based LM with different filler models. In all LMs n was set to 3. It must be noted that the DI class-based LM has served as the LM of the baseline system with 921 unigrams including 19 classes. The total number of the distinct words in the lexicon was 1681. The 48 semantic grammars with fillers were designed so to cover the lexicon.

The first set of experiments were carried out using different filler models. The results are presented in Table 1. Although the differences are not significant the D_POS filler model yields the best performance. We think this model is a good tradeoff between model resolution and data sparseness. The second set of experiments were performed by interpolating the grammar based LM with word and class/word LMs. The filler model was D_POS.The results are presented in Table 2. The best system has turned out to be the system with the class/word interpolation. The real time performance of the best system has been found 4% worse than the baseline system. This clearly illustrates that the word graphs and in turn the concept graphs are very compact in our task, and that our top-down chart based partial parser with a fairly small number of semantic grammars is very efficient.

6. Future work

Motivated by the results presented in the preceding section we started to look at a tighter implementation of the speech understanding framework introduced in Section 2. The generative model based on our framework will be the basis for the search network in our future work. The generative model of spoken utterances is shown in Figure 5 for the bigram concept LM for the sake of simplicity. In fact, the generative model defines the search network for the MAP optimization (see equation (4)) explained in the preceding section. So, the search network is a linear structure of concepts with a background model that accounts for segments of speech waveform not covered by the concepts. The dashed lines indicate the recursive nature of the concept labeled SCFGs (or SRTNs). In this tighter interface we have concepts, $c_1, c_2 \cdots$, compiled into extended SRTNs (ESRTNs); each terminal arc is extended using the respective word HMM model. We suggest dialog context dependent 3-gram modeling of the concept sequence as in the word graph interface. The network can be searched in either top-down or bottom-up manner. The former probably needs an active chart based agenda driven parser [3] whereas the latter can be implemented using the multi-token passing paradigm with Viterbi recombination [2]. At the present, we do not know which approach will result in a computationally more efficient system. It must be noted that a hybrid implementation of both approaches is also possible.

Table 2: Word error rate results of different interpolation methods

Method	WER	Relative gain
Baseline	22.8%	0.0%
grammar LM alone	22.0%	3.5%
word+grammar LM	21.7%	4.8%
class/word+grammar LM	21.1%	7.5%

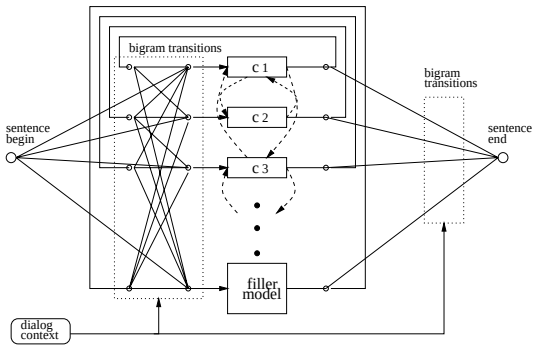

Figure 5: The generative model that can also be used as a search network.

7. References

[1] M. K. Ravishankar, *Efficient Algorithms for Speech Recognition*, Ph.D. thesis, Carnegie Mellon University, 1996.

[2] K. Hacioglu, "A decoding algorithm based on word lattice and multi-token passing paradigm," *unpublished work*, Center for Spoken Language Research, 2000.

[3] H. Weber, "Time synchronous chart parsing of speech integrating unification grammars with statistics," in *Proceedings of Twente Workshop on Speech and Language Engineering*, December 1994, pp. 107–120.

[4] K. Hacioglu and W. Ward, "Dialog-context dependent language models combining n-grams and stochastic context-free grammars," in *International Conference of Acoustics, Speech, and Signal Processing*, Salt-Lake, Utah,, 2001.

[5] K. Hacioglu and W. Ward, "On combining language models: Oracle approach," in *First International Conference on Human Language Technology Research*, San Diego, California,, March 18-21 2001.

[6] W. Ward and B. Pellom, "The CU communicator system," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, Keystone, Colorado, 1999.

[7] W. Ward, "The cmu air travel information service: Understanding spontaneous speech," in *Proceedings of the DARPA Speech and Natural Language Workshop*, June 1999, pp. 127–129.

[8] B. Souvignier, A. Keller, B. Rueber, H.Schramm, and F. Seide, "The thoughtful elephant: Strategies for spoken dialog systems," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 1, pp. 51–62, January 2000.

[9] A. Kellner S. C. Martin and T. Portele, "Interpolation of stochastic grammar and word bigram models in natural language understanding," in *6-th International Conference on Spoken Language Processing*, Beijing, China, 2000, pp. 1695–1699.

[10] A. Keller, B. Rueber, F. Seide, and B.H. Tran, "PADIS - an automatic telephone switchboard and directory information system," *Speech Communication*, vol. 23, pp. 95–111, 1997.