EFFICIENT GRAMMAR GENERATION AND TUNING FOR INTERACTIVE VOICE RESPONSE APPLICATIONS

Ellis K. Cave

Intervoice Inc.
Dallas, Texas 75252 USA
Skip.Cave@intervoice.com

Mithun Balakrishna and Dan Moldovan

The University of Texas at Dallas
Richardson, Texas 75080 USA
{mithun,moldovan}@hlt.utdallas.edu

ABSTRACT

This paper presents a procedure to efficiently create and tune context free grammars for directed dialog speech applications using only spoken test user utterances. We present a procedure to transcribe utterances with improved accuracy by post-processing the ASR n-best lists with higher level knowledge sources and additional information from the application prompt. We then present a semantic categorizer for the transcriptions, a statistical filtering mechanism for modifying the grammars and, a mechanism to raise an alarm condition in case of large in-flow of errors. We also illustrate the importance of additional improvements gained by using the semantic classification strength in a feedback loop to the transcription mechanism.

1. INTRODUCTION

The current generation of telephone based directed dialog speech applications (DDSAs) predominantly use context free grammar (CFG) instead of a n-gram based language model (LM)[1]. The preference for CFG in telephonic Interactive Voice Response (IVR) systems can be attributed to the very tight constraint placed on the ASR's response time to a user's request and the limited availability of text corpora for a wide range of application domains. The need for only the accurate semantic tag and its corresponding arguments associated with the user response rather than the entire set of words spoken by the user also justifies this preference.

In the conventional grammar based IVR, the user response to a particular IVR prompt is fed to a CFG based ASR. The ASR uses the CFGs to decide whether the user response is valid (Match) or invalid (No-Match) and the IVR's response is based on this decision. For the prompt "do you want your account balance or cleared checks?", a CFG might accept replies with words like "checks" or "balance." If the user responds "what is the price of pizza?," the system categorizes it as a no-match and repeats the prompt. Every user prompt in a DDSA needs a CFG to accurately recognize and semantically classify the user's response for that particular prompt. For the IVR to work with maximum accuracy, the IVR CFGs should cover the most probable responses that are expected from the user at every prompt in the application call-flow [1]. The success of well-designed CFGs has resulted in the very negligible deployment of their statistical n-gram LM based counterparts.

Reference [2] proposes a semantically structured model, containing a combination of statistical n-grams and CFGs, to reduce the manual labor in developing CFGs. The proposed method however requires a partially labeled (manually performed) text corpus in the IVR's domain for training the semantically structured model. Due to the high deployment demand for directed dialog systems in a wide variety of domains and the lack of any respectable size text corpora in these areas, the traditional manual processes (performed by a qualified speech application designer) of creating and tuning CFGs use only the collected test user speech utterances. Call-routing algorithms [3] have been proposed to deal with the IVR grammar generation problems but CFGs are still the best models for command-and-control scenarios where the objective is to map the user utterance into a particular command possibly with slots or variables.

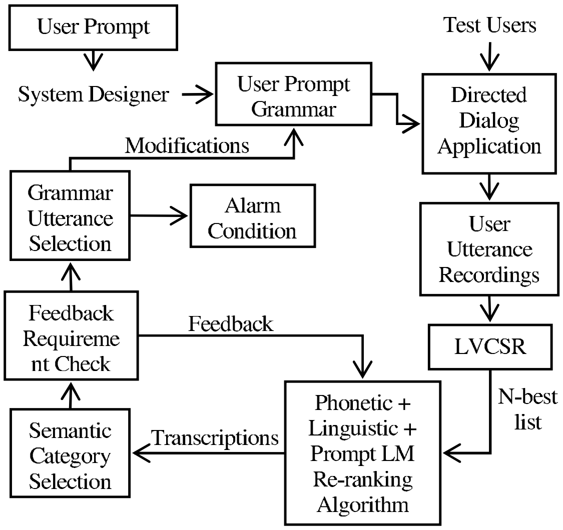
User Prompt
Test Users
System Designer
User Prompt Grammar
Directed Dialog Application
Modifications
Grammar Utterance Selection
Alarm Condition
User Utterance Recordings
Feedback Requireme nt Check
Feedback
LVCSR
N-best list
Semantic Category Selection
Transcriptions
Phonetic + Linguistic + Prompt LM Re-ranking Algorithm

Fig. 1. Automatic creation and tuning of CFG

The main purpose of this paper is to describe a method of automatically creating and tuning grammars representing users' responses to prompts in IVR systems. For example, for the prompt "do you want your account balance or your cleared checks?", users might reply, "Total in my account" or "Sum of my account"; and our goal is to produce an efficient grammar containing the full set of actual responses. It should be noted that the goal is not to produce a grammar containing all the possible user response alternatives to a particular prompt but to produce one containing only the most probable user responses required to improve the IVR performance.

2. AUTOCFGPROCESSOR

Fig. 1 depicts our proposed design. The goal of the system is to capture the most probable user responses, which have not yet been adequately represented by the semantic tags (creation task) or the previously created CFG (tuning task). A set of test callers respond to the IVR loaded with the speech designer conceived prompts, semantic tags and previously created CFGs (only for CFG tuning). “Wizard-of-oz” procedure or a skeletal CFG is used to guide the callers through the call flow. The user responses are recorded and transcribed using a Large Vocabulary Continuous Speech Recognizer (LVCSR) system with a reranking mechanism and a prompt based LM. The WordNet [4] based semantic categorizer then tries to map the user utterance transcription to the best semantic label or previous CFG's response alternatives (only for CFG tuning). A multiple loop feedback mechanism between the semantic categorizer and reranking module is used to improve the transcription accuracy.

The (utterance transcription, semantic label) pairs are statistically validated by a selection mechanism to not only add some valid alternatives but also remove some statistically invalid entries from the old CFG (we have observed that bigger CFGs do not necessarily lead to a better IVR performance, as more alternatives increase the ASR confusion). The created/tuned CFG is loaded into the IVR and the entire process is repeated until no further improvement is achieved. The selection mechanism also determines, based on the constructed statistical base, if an alarm should be raised. The system might find a statistically significant number of similar responses to a prompt which cannot be mapped to any semantic category of the IVR prompt. For example, if a bank sends its customers improper account statements, the customers will call-in and, the prompt “do you want your account balance” will generate responses like “the statement is incorrect” or “I have a wrong statement”. These responses certainly cannot be anticipated and thus will be not in the grammar. In such situations, it is prudent that we raise an alarm and control any further damage to the external environment.

2.1. N-Best List Reranking Based Transcription Process

Since our corpus of user responses is spoken, a necessary first step in the automation process is transcribing the responses more accurately than has previously been possible. Experimental results [5] prove that considerable improvements can be gained by applying a strong reranking mechanism, even at a small n-best list depth. Hence, we transcribe the test user utterances with considerable accuracy using a LVCSR along with the n-best list reranking mechanism presented in [5], modified with additional extensive phonetic and semantic knowledge and, a applications prompt based LM. The reranking score assigned to the LVCSR n-best hypotheses is a simple linear weighed combination of the individual scores from the domain independent phonetic, lexical, syntactic and semantic knowledge sources. The reranking mechanism makes them work in tandem, complementing each other to improve the WER.

For every n-best list hypothesis, the confidence score is computed in the following manner: if $Ph = ph_1, ph_2, \ldots, ph_m$ is the phoneme sequence corresponding to the n-best list hypothesis word sequence $W_h = w_1, w_2, \ldots, w_k$, predicted by the LVCSR for the acoustic frames A of an utterance, then

$$\begin{aligned}
Score\left(W_h\right) = &\\
& w_1 * \left(P\left(Cat\left(Ph\right)|A\right) * P\left(Ph|Cat\left(Ph\right) \wedge A\right)\right)\\
& + w_2 * \frac{1 - P\left(Ph^s|Ph \wedge A\right)}{1 - P\left(Ph^s|Ph\right)}\\
& + w_3 * \left(P_{LM}\left(W_h\right) * \prod_{i=1}^{k} \frac{P\left(w_h^i \mid B\left(w_h^i\right)\right)}{P\left(W \mid B\left(w_h^i\right)\right)}\right)\\
& + w_4 * \left(P\left(\pi, W_h\right)\right)\\
& + w_5 * \left(\sum_{i=1}^{r} \frac{\frac{\sum_{j=1}^{p} Cnt(FillSlot(j,W_h,v_i) \wedge v_i)}{\sum_{j=1}^{q} Cnt(FillSlot(j,v_i) \wedge v_i)}}{\frac{Cnt(v_i)}{\sum_{j=1}^{t} Cnt\left(v_j\right)}}\right)\\
& + w_6 * LCS\left(W_h\right) + w_7 * Prompt\_LM\left(W_h\right) \qquad (1)
\end{aligned}$$

1. Grammar: Cable Account Change
2. User Utterance: “I'd like to speak to a live person please”
3. LVCSR Sequence **(LVCSR-S)***: “I'd like/VB to speak/VB to a live/JJ person/NN of these”*
4. Target Semantic Tag **(TST)***: “Customer Service”*
5. Target Grammar Entry **(TGE)***: “Human/JJ/#2 Operator /NN/#2”*
6. Best 2 *Lexical Chains of* 8*:*
a. operator/NN/#2 TO speak/VB: (n−operator#2, manipulator#1) HYPONYM (n−telephone_operator#1, telephonist#1, switchboard_operator#1) GLOSS (v−get#14) HYPERNYM (v−communicate#2, intercommunicate#2) HYPONYM (v−talk#1, speak#2) VALUE: 9.22
b. human/JJ/#2 TO person/NN: (a−human#2) GLOSS (n−person#1, individual#1, someone#1, somebody#1, mortal#1, human#1, soul#2) VALUE: 6.68
7. Lexical Chain Strength **(LCS)***:* 15.90

Fig. 2. An illustrating of mapping in semantic categorization

In (1), $(w_1, w_2), w_3, w_4, w_5$ represent the weights assigned to the phonetic, lexical, syntactic and semantic features respectively. Reference [5] gives a detailed description of the constituents of the reranking equation in (1) and an algorithm to produce a better hypothesis by reranking the nbest list with these weighted knowledge sources. Additionally, w_6 is the weight assigned to the lexical chain strength of the hypothesis and will be explained in the next section. w_7 is the weight assigned to the prompt based LM score for W_h. The prompt based LM is created from the IVR prompt utterance transcription, semantic categories and prompt grammar (only for CFG tuning). Basically, we are trying to give the LVCSR hypothesis (got without any prior knowledge of the domain), which contain the words present in the prompt LM, a higher score when re-ranking. This will not result in complete elimination of out-of-grammar utterances because the LVCSR is configured for open domain utterances.

2.2. Semantic Categorizer and Feedback

Ref. [6] present a methodology for finding topically related words by increasing the connectivity between WordNet [4] synsets using the information from WordNet glosses. Thus, we can find if a pair of words are closely related by not only looking at the WordNet synsets but also by finding lexical paths between the word pair using the WordNet synsets and glosses. Hence, we use lexical chains to classify the LVCSR-transcription into one of the designer conceived semantic tags. The semantic categorizer tries to map the user utterance transcription to the best semantic label or their corresponding response alternatives from the previous CFG (only for CFG tuning).

Fig. 2 illustrates the mapping process. Lexical chains are found between each content word in the LVCSR-S and each content word in the TGE (semantic label or, word sequences from the previously created CFG representing the TST). Prior to the mapping process, all TGE words are manually assigned their POS tags and WordNet2.0 sense numbers. The LVCSR-S words are not sense disambiguated. A LVCSR-S to TGE mapping is *valid* if and only if there exists a lexical chain between every word in TGE and at least one word in

```
For each A in utterances of Tuning Task
  For each B in top ten LVCSR-S of n-best
    For each C in TSTs of Tuning Task
      For each D in TGEs of C
        If ValidMapping(B,D)
          BestLexicalChain(B,D,LCS[B,D])
        else
          LCS[B,D] = NO-MATCH
      Value[C] = TGE_Max(LCS[B,D])
    TST[B] = TST_Max(C,Value[C])
  Sem_Cat(A) = Majority(TST[B])
```

Fig. 3. Algorithm to map the LVCSR n-best list transcriptions into the best semantic category for the IVR prompt

the LVCSR-S. The LCS is the sum of the semantic similarity values of the best lexical chains from every TGE word.

In our example, the LVCSR-S to TGE mapping is valid because *human* and *operator* map to at least one word in LVCSR-S (*person* and *speak*). The LCS is 15.90 (sum of best lexical chain values) i.e (*operator* to *speak*, Value= 9.22) and (*human* to *person*, Value= 6.68). The first lexical chain (a) links the second WordNet2.0 sense of the noun *operator* with the verb *speak*. The lexical chain implementation [6] does not require a word sense for *speak*. It finds the best path to the second WordNet2.0 sense of *speak* and assigns a value of 9.22 to the found path (higher values indicate stronger paths) and, hence we find the strongest word pair mappings for the content words in the LVCSR and TGE.

Fig. 3 presents the algorithm to map an user utterance transcription into the best semantic category for each prompt. For each user utterance, the top ten LVCSR transcriptions are selected and mapped to one of the various semantic categories available for the prompt using the previously illustrated lexical chain mechanism. A majority voting mechanism is then used to find the best semantic category from the various semantic categories proposed by the top ten LVCSR transcriptions. The confidence score associated with the semantic category chosen from the majority voting procedure is used to rerank the top ten hypotheses again and the process continues until the algorithm settles down to a particular transcription and category.

The procedure `ValidMapping()` identifies the mapping between a given (LVCSR-S, TGE) pair and returns `true` if and only if the validity condition for mapping holds. For such a valid pair, `BestLexicalChain()` computes the LCS. `TGE_Max()` finds the best TGE (based on LCS) for every TST and `TST_Max()` returns the best TGT (based on the LCS of the best TGE) for the top ten LVCSR-S of the n-best list. `Majority()` selects the TGT with the majority LVCSR-S votes as the utterance semantic category. The selection of the word sequence representing the TST is as important as the correct TST selection. We use the LCS of a hypothesis, associated with the chosen TGT for the utterance, as a confidence score to rerank the top ten hypotheses again and the process continues until the system settles down to a transcription and semantic label pair.

There are some transcription words which are currently unavailable in WordNet2.0. These missing transcription words cannot take part in the semantic categorization process as the lexical chains procedure relies on the availability of the word in WordNet2.0 to map it to the previously prepared descriptions of possible responses. In this paper, the missing transcription words are ignored since we have found their frequency of occurrence in our actual user utterance transcriptions to be negligible.

2.3. Statistical Selection Mechanism

We propose an statistical selection mechanism to not only add user response alternatives but also remove dormant word sequences from the CFGs to improve the IVR performance by filtering (transcription, semantic category) pairs using the rules below:

Occurrence Probability: Each response alternative inside the CFG contains a value indicating the occurrence probability of the word sequence for a particular semantic category in each prompt. This value indicates the probability of the word sequence being uttered by the user for that semantic category. We prune the CFG to prevent the degradation in the IVR performance due to presence of low probability word sequences previously added into the CFG. We remove response alternatives with occurrence probability lower than 0.05 and then re-adjust the occurrence probabilities of all the remaining alternatives for that particular prompt semantic category (occurrence probabilities sum of all the alternatives for a semantic category is 1). This rule is applied once to remove low probability response alternatives after each of the below rules which add valid LVCSR-transcriptions into the CFG .

Frequency: For a particular semantic category, add only widely used LVCSR-transcriptions (frequency greater than 1% of total valid LVCSR-transcriptions) e.g. a specific transcription like “Account number five one four three” should not be added in its entirety, though the user wants the IVR to map the account to a semantic category (Cable, Cellular Phone, etc.) in the prompt. We then recalculate the occurrence probability of all the response alternatives.

Smallest Sequence: If a LVCSR-transcription S_1 is a sub-string of S_2 (both with a frequency greater than the threshold and both mapped to the same semantic tag), then only the smaller sequence S_1 should be added to the CFG. e.g. If $S_1 =$ “billing address” and $S_2 =$ “get my billing address,” add S_1 to the CFG. The occurrence probability of all the response alternatives is recalculated.

Sub-sequences: We need to consider the fact that the entire transcription might not be valid due to LVCSR errors or invalid user words. We might have low frequency LVCSR-transcriptions containing important, high frequency word sub-sequences e.g. the utterance “Ah Can I just speak to someone alive” is transcribed by the LVCSR as “that called religious speak to someone by.” We find valid LVCSR transcription sub-sequences by extracting the words used to compute the LCS, for its corresponding semantic tag, in each low frequency LVCSR-transcription. If the count of a particular sub-sequence in a semantic tag class is greater than the frequency threshold, then the corresponding sub-sequence is added to the CFG for that semantic tag. We are basically trying to filter the worthless words and find useful high frequency sub-sequences. The occurrence probability of all the response alternatives is recalculated with the count of the added sub-sequence.

2.4. Alarm Condition

We need to determine based on the invalid utterances set, if an alarm should be raised. The steps from the section 2.3 can be repeated for invalid utterances and an alarm is raised if similar transcriptions have a frequency greater than the threshold. In the previous example from section 2 about customers calling in about incorrect bank statements, the number of varied responses from the callers will result in a large number of null semantic category responses with lexically dissimilar transcriptions. Hence, we need to devise a more robust mechanism to detect the alarm conditions. In every iteration of the grammar tuning, for a given prompt, if the number of null semantic category responses are greater than a percentage of the valid responses (usually set to 25%), then we run the semantic categorizer procedure for the invalid

Table 1. Various WER results obtained for the AUTOCFGPROCESSOR transcription task.

w_1=*13*,w_2=*18*,w_3=*14* w_4=*26*,w_5=*29*	Test User Utterance Set (8013 Utterances)				
	Error (%)				Total Correct(%)
	Sub	Del	Ins	Total	
Baseline Hypothesis	29.7	7.9	10.1	**47.7**	**62.4**
30-Best List Reranking	25.5	5.1	10.2	**40.8**	**69.4**
30-Best List Reranking + LCS Score Feedback + Prompt based LM	24.4	4.0	8.4	**36.8**	**71.6**

utterances. Initially, all invalid utterances are placed in their own individual clusters. Each cluster is then compared against another to check if they can be merged based on the lexical chain strength. After all the clusters have been compared, an alarm is raised if the largest cluster of invalid responses is greater than the threshold.

3. RESULTS AND EXPERIMENTAL SETTINGS

We use SONIC [7], a LVCSR system from the University of Colorado at Boulder, to produce n-best lists for the reranking mechanism. We trained the acoustic model for the telephone transcription task using 160 CallHome and 4826 Switchboard-1 conversation sides. The SRI HUB5 2000 model is used as a back-off tri-gram LM.

We collected a set of 8013 user utterances (live Intervoice Inc. IVR recordings) for 4 prompts in 3 different IVRs: *Change Cable Account Details* (CCAD) (2452 utterances, 12 semantic categories), *Billing Account Type Choice* (BATC) (1328 utterances, 12 semantic categories), *Bank Future Payment Update Options* (BFPUO) (224 utterances, 9 semantic categories), *Wireless Account Change Options* (WACO) (4009 utterances, 13 semantic categories).

Table 1 presents the transcription results obtained on a 30-best list reranking using the best set of knowledge weights obtained by running WER testing trials on 40 HUB5-2000 Switchboard conversations using various weight combinations. Using the best weight combination, we achieve 6.9% absolute WER reduction (14.47% relative reduction). We got the best result (10.9% absolute WER reduction and 22.85% relative WER reduction) by using the recurring LCS score feedback ($w_6 = 72$) and the application prompt information based LM ($w_7 = 28$) to propose the best hypothesis from the reranked top 10 n-best list hypotheses.

The CFG tuning performance is presented in Table 3 with the manually created CFGs for the 4 prompts as the baseline. An application designer manually listed the semantic categories for each prompt. The 8013 utterances set is divided equally, for each prompt, into a training and test set. *Original* system results are obtained by running the test set using the original manually created CFGs. The system results (*Manual*, *Auto1* and *Auto2*) represent the quality of the prompt CFGs (tuned on the training set) against the test set. *Manual* system prompt CFGs are obtained by tuning the original CFGs with word sequences proposed by a speech application designer, who manually analyzed the training set. *Auto2* (AUTOCFGPROCESSOR) system tunes the original CFGs with the word sequences obtained from the LVCSR reranking, categorization & feedback and, statistical selection while *Auto1* is the implementation of [5].

MisCat errors are due to mismatches between the category proposed by the IVR and the actual utterance category. *InCFG* errors are due to the IVR proposing a category while the utterance's actual category is a NO-MATCH. *OutCFG* errors are due to the IVR proposing a NO-MATCH while the utterance actually has a valid category. Table 3 shows that the AUTOCFGPROCESSOR can successfully add good alternatives to the baseline CFG and improve the IVR performance. *Auto1* consistently comes close to matching the performance of *manual* additions, though it adds more alternatives and hence achieves less *MisCat* errors and more *InCFG* errors than the manually tuned CFGs. Using the statistical selection mechanism to remove dormant word sequences from the CFGs, *Auto2* improves the performance of the IVR even in terms of *InCFG* errors. The multiple loop feedback and the prompt based LM presents better alternatives and hence, reduces the *MisCat* errors.

Table 2. Results obtained for the AUTOCFGPROCESSOR directed dialog speech application task.

Prompt (Size)	System	Collected Test User Utterance Set (4006 Utterances)				
		Error (%)				Total(%) Correct
		MisCat	InCFG	OutCFG	Total	
CCAD (1226)	Original	6.53	7.91	3.92	**18.35**	**81.65**
	Manual	4.57	8.24	2.12	**14.92**	**85.07**
	Auto1	4.15	8.48	2.45	**15.08**	**84.91**
	Auto2	4.16	8.32	2.12	**14.6**	**85.40**
BATC (664)	Original	4.07	4.52	4.07	**12.65**	**87.35**
	Manual	3.16	4.36	3.31	**10.84**	**89.16**
	Auto1	3.46	4.52	3.61	**11.60**	**88.40**
	Auto2	3.01	4.52	3.61	**11.14**	**88.86**
BFPUO (112)	Original	3.57	4.46	6.25	**14.29**	**85.71**
	Manual	3.57	4.46	6.25	**14.29**	**85.71**
	Auto1	2.68	5.36	7.14	**15.18**	**84.82**
	Auto2	1.78	3.57	7.14	**12.5**	**87.50**
WACO (2004)	Original	6.24	8.98	5.04	**20.26**	**79.74**
	Manual	2.74	10.18	2.05	**14.97**	**85.03**
	Auto1	2.4	10.33	3.34	**16.07**	**83.93**
	Auto2	2.2	9.98	3.04	**15.22**	**84.78**

4. CONCLUSIONS AND FUTURE WORK

This paper presented a novel method to automatically generate and tune grammars for IVRs. The results show that the improved IVR performance closely matching the manual tuning performance. Non-domain specific information sources based reranking, the application prompt information based LM, lexical chain based semantic category classification and, feedback based on semantic classification strength play an important role in improving the IVR performance.

5. REFERENCES

[1] Intervoice Inc., *Intervoice Training Document - Voice User Interface Design - Speechworks 7.0 OSS/OSR and Naunce 8.0 - Speech Forms*, 2004.

[2] A. Acero, Y. Y. Wang, and K. Wang, "A semantically structured language model," in *Special Workshop in Maui (SWIM)*, 2004.

[3] Q. Huang and S. Cox, "Automatic call-routing without transcriptions," in *Eurospeech*, 2003.

[4] C. Fellbaum, Ed., *WordNet: An Electronic Lexical Database*, MIT Press, 1998.

[5] M. Balakrishna, D. Moldovan, and E. K. Cave, "Higher level phonetic and linguistic knowledge to improve asr accuracy and its relevance in interactive voice response systems," in *AAAI Workshop on SLU*, 2005.

[6] D. Moldovan and A. Novischi, "Lexical chains for question answering," in *COLING*, 2002.

[7] B. Pellom, *SONIC: The University of Colorado Continuous Speech Recognizer*, University of Colorado, May 2005.