Automatic Generation of Statistical Language Models for Interactive Voice Response Applications

Mithun Balakrishna, Cyril Cerovic, Dan Moldovan

Human Language Technology Research Institute
The University of Texas at Dallas, Texas, USA
{mithun,ccerovic,moldovan}@hlt.utdallas.edu

Ellis Cave

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

Abstract

This paper proposes a methodology to automatically generate statistical language models (SLM)s for the recognition of utterances in Interactive Voice Response (IVR) systems. The paper aims at creating SLMs for each IVR prompt [1] with minimum amount of human intervention and prior knowledge regarding the expected user utterances at a particular prompt. A combination of prefiller patterns based on spontaneous speech utterances, WordNet [2] and Roget's thesaurus based content word extraction and, world wide web based statistical validation is used to generate SLMs automatically. The created SLM not only reduces the manual labor involved in IVR application development but also focuses on minimizing the Word Error Rate (WER) and the Semantic Error Rate (SemER) of the ASR transcriptions. We use a WordNet [2] lexical chain based semantic categorizer to classify ASR transcriptions into semantic categories representing each IVR prompt.

Index Terms: automatic speech recognition, statistical language model, interactive voice response systems, semantic categorizer.

1. Introduction

The current generation of telephone based Directed Dialog Speech Applications (DDSAs) predominantly use Context Free Grammars (CFGs) instead of n-gram based statistical language models (SLMs) [1]. The preference for CFGs in telephonic Interactive Voice Response (IVR) systems can be attributed to the limited availability of text corpora to train good quality SLMs for various domains. This preference is also justified by the need for accurate semantic tags and arguments rather than low transcription Word Error Rate (WER). A CFG in an IVR call-flow prompt contains universals (e.g. "help," "operator") and a set of most probable words or phrases expected from the user at that prompt [1].

The IVR accuracy is directly dependent on the CFGs' coverage of the expected user responses at every prompt [1]. The success of well-designed CFGs has resulted in a very negligible deployment of their SLM based counterparts. Still, the CFGs are manual labor intensive and suffer from the lack of coverage. The CFGs place a very tight constraint on the users' response to a particular prompt and regard any variation of the expected responses as a "no-match". For example, at the prompt "do you want your account balance or cleared checks?", a CFG accepts replies with words "checks" or "balance" but rejects responses such as "account sum" or "account total". Since the CFG creation process is predominantly manual, it requires a huge amount of effort by a qualified speech application designer to produce an IVR with a decent SemER (a measure of the errors in the IVR proposed semantic categories). It is also very difficult to represent the probability distribution for the various response alternatives.

Reference [3] proposed 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 model training. Call-routing algorithms [4] have been proposed to deal with the IVR CFG/SLM generation problems but they still require a set of speech utterances for the application domain and; CFGs and SLMs are still the best models for command-and-control scenarios to map transcriptions to commands with slots or variables. References [5, 6] proposed a WordNet lexical chain [7] based methodology to efficiently create CFGs and subsequently tune them. These methods again require the collection of a decent sized speech utterance set which is an expensive process.

References [8, 9] proposed a methodology to combine World Wide Web (WWW) based multiple text sources to train SLMs for the conversational speech task. Taking inspiration from the success of these techniques, we propose a methodology combining spontaneous speech utterance based prefiller patterns, WordNet [2] and Roget's thesaurus based content word extraction and, WWW based statistical validation to generate SLMs automatically. Our proposed methodology requires a minimum amount of human intervention and no prior knowledge (text corpora, user utterance collection or manually created CFGs) regarding the expected user utterances at a particular prompt. For evaluation, we use a WordNet [2] lexical chain based semantic categorizer to classify ASR transcriptions into semantic categories and compare these semantic categories against the manually labeled utterance categories. The goal of the paper is to generate SLMs automatically, not only to reduce manual labor involved in IVR application creation but also to reduce the SemER of these IVR applications.

2. Automatic SLM generation

This section presents techniques which work in tandem to automatically generate SLMs with minimum manual intervention and, without any text corpora, user utterance collection or manually created CFGs for the IVR domain. To produce the SLM for a particular IVR prompt, we need to provide semantic category labels, a description for each one of these labels and the possible task labels defined by the IVR prompt for each semantic category. Table. 1 presents an example of the input requirements to generate the SLM for the "Account Payment" prompt.

Fig. 1 depicts our proposed automatic SLM generation design.

Table 1: SLM input requirement for "Account Payment" prompt.

Semantic Category	Description	Task Label(s)
arrange_a_payment	users can arrange payments	arrange a payment
report_a_payment	users can report previously made payments	report a payment
payment_methods	users can hear about payment methods and other payment options	hear payment methods
billing_information	users can hear about their billing information or check their account balance	hear complete billing information, check account balance
credit_card_payment	users can make a credit card payment	make a credit card payment

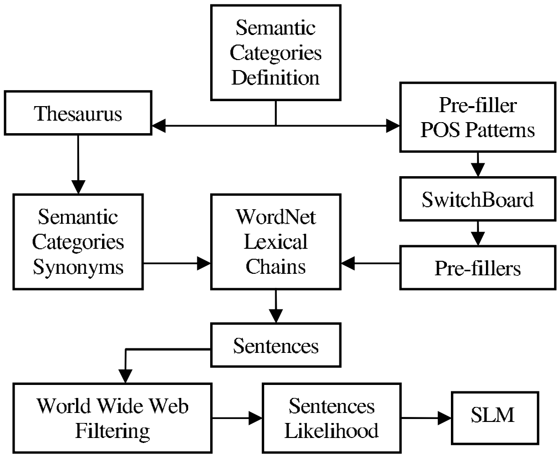

Figure 1: Proposed methodology to automatically generate SLMs.

The three main modules involved in the SLM generation are: spontaneous speech utterance based prefiller pattern extraction, WordNet [2] and Roget's thesaurus based content word extraction and, WWW and WordNet based statistical validation mechanism. Each valid user utterance can be broken into 3 parts: prefiller words, content words and post-filler words. However, prefiller words and content words constitute the majority of the utterance transcription words and have the biggest influence on the WER and SemER.

2.1. Extracting prefiller words

We use each semantic category description to extract certain prefiller part-of-speech (POS) patterns. Table. 2 presents some POS patterns extracted to represent the prefiller words that can be uttered by the user for that particular semantic category. After the manual extraction of POS patterns from 20 semantic category descriptions, we observed that the prefiller POS patterns for these semantic categories are picked from a pool of 8 POS patterns. Hence, we decided to use the identified pool of patterns for all the remaining semantic category descriptions (to keep the manual labor to a minimum) and let the filtering modules deal with non-compatibility of the generalized POS patterns with certain semantic categories. The POS patterns from the pool are then searched for in 1126 POS tagged SwitchBoard-1 conversations from the TreeBank-3 corpus to extract spontaneous/conversational speech style prefiller words.

Table 2: Prefiller words extracted for some POS patterns.

Category & Description	POS Pattern & Example Utterance
Cable_Account - Users want to check their cable account bill	prp vb nn - I want NN, I need NN vb prp nn - check my NN, give me NN vb nn - pay NN prp vb vb nn - I'd like to have NN prp vp prp nn - (can) you give me NN

Table 3: Extracted content word alternatives for a sample category.

Category & Description	Content Words & Alternatives
Cellular_Phone - Users want to check their cellular phone bill	car telephone, cell phone, cell telephone, cellular phone, digital telephone, field telephone, satellite telephone, wireless telephone

The search module presents 3 different prefiller word information: Pure POS Pattern Prefiller Words - identified prefiller words which strictly adhere to the POS patterns e.g. "I want credit" for the pattern "PRP VB NN", POS Pattern Prefiller Words with Gaps - identified prefiller words which comply with the POS patterns but with some gaps between POS tags in the pattern e.g. "I want to get another brand" for the pattern "PRP VB VB NN", POS Pattern Prefiller Words with Additional Peripheral Words - identified prefiller words for "Pure POS Pattern Prefiller Words" and "POS Pattern Prefiller Words with Gaps" but some additional peripheral words in the beginning and end of the POS pattern e.g. "Could I have something" for the pattern "PRP VB NN". We remove the "NN" words from all the identified prefiller words and we replace all the "PRP" words with appropriate personal or possessive pronouns depending on the POS pattern e.g. "PRP" words for the pattern "PRP VB NN" are replaced by "I" and "we".

2.2. Extracting content words

For each semantic category, we use its description to extract a skeletal set of content words. We then use the Roget's thesaurus to find a set of alternatives closely related to these sets of content words. Table. 3 presents the word alternatives extracted for the category "cellular_phone". The output from the thesaurus contains good alternatives for the content words, however they contain irrelevant words too e.g. for the category "arrange_a_payment", we find the alternatives by combining the closely related words for "arrange" and "payment" and this leads to some noisy alternatives like "adapt deposit" or "organize fee".

Reference [7] presents a methodology for finding topically related words by increasing the connectivity between WordNet [2] 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. To remove the noisy alternatives, we use the WordNet based lexical chains [7] to find a connection between the words present in the alternatives e.g. the lexical chain between the words "adapt" and "deposit" has a low confidence score of 1.2026, while the word pair "prepare" and "amount" has a relatively higher confidence score of 14.5549. Hence, an alternative is considered to be valid and is added to the list only if the lexical chain confidence score for its content words is greater than a threshold value. In summary, after the completion of these steps, a set of possible prefiller and content words

```
For each utterance transcription A
  For each semantic category B in prompt
    For each word alternative C for B
      If ValidMapping(A,C)
        BestLexicalChain(A,C) = Max(LCS(A,C))
      If Value(A,B) < BestLexicalChain(A,C)
        Value(A,B) = BestLexicalChain(A,C)
    Sem_Cat(A) = Decay_Threshold(Value(A,B))
```

Figure 2: Algorithm to semantically categorize transcriptions.

representing each IVR prompt is collected.

2.3. Statistical validation

Next, we try to combine each prefiller phrase with every content word phrase to form a complete word alternative. A particular prefiller phrase is combined with a content word phrase only if we find a WordNet lexical chain between the prefiller phrase verb and the noun/verb in the content word phrase (if it is a noun phrase/verb phrase) with a confidence score greater than a defined threshold. The lexical chain confidence score for a word pair is usually determined by the presence of one word in the WordNet gloss of the other word and vice-versa. The lengthier the chain i.e. extending to the glosses and reverse-glosses of the hyponymns or hypernymns for the word pair, the smaller is the lexical chain confidence score. The complete sentence thus formed is then filtered using a WWW based statistical validation mechanism. We use the google search engine to search for the new sentences on the web (also on news groups since they are close to conversational style text) as one cohesive unit. If the count (number of web page links) returned by the search engine exceeds a defined threshold then the sentence is added to the data set later used to build the SLM. The count provided by the web for a particular alternative is also used to represent its probability distribution in the SLM data set.

3. Semantic categorizer

In order to evaluate our SLMs, we use a WordNet [2] lexical chain based semantic categorizer first proposed in [5, 6] to classify the ASR transcriptions into one of the semantic tags. In this paper, we modify and extend this categorizer to map the ASR transcription into multiple semantic categories. Semantic grammar based categorization methods and statistical categorization methods cannot be used due to the non-availability of hand written semantic rules or any training text corpora in the IVR application domain.

Fig. 2 presents our semantic categorization algorithm. Words in each transcription and all the word alternatives for a semantic category are assigned a POS tag using the Brill's tagger and a WordNet word sense using an in-house system for Word Sense Disambiguation (WSD) of open text. The procedure `ValidMapping()` then identifies the mapping between a given transcription and the semantic category's word alternative. It returns `true` only if there exists a lexical chain between every word in the word alternative and at least one transcription word. The LCS is the sum of the semantic similarity values for the best lexical chains from every word in the alternative to a word in the transcription. We then identify the best LCS for such a valid (transcription, word alternative) pair. Each semantic category is then assigned the best LCS value from all its word alternatives. A transcription is assigned to a semantic category if the LCS value of the transcription for that semantic category is greater than an absolute LCS threshold value. To allow a transcription to map to more than one semantic category, we define a LCS difference threshold value. Hence, any transcription is first mapped to the best semantic category (with the highest LCS value which is greater than the absolute LCS threshold value) and, to any other semantic category (with a LCS value $> Max(($LCS value of the best semantic category $-$ LCS difference threshold value), absolute LCS threshold value)). We also define a LCS difference decaying factor, which is the factor used to reduce the LCS difference threshold value as the number of semantic categories assigned to a transcription grows.

Table 4: Transcription WER results obtained for the test set.

	Test User Utterance Set (20804 Utterances)				
	Error (%)				Total
	Sub	Del	Ins	Total	Correct(%)
Oracle-SLM	5.2	4.5	5.3	**15.0**	**90.3**
Nuance-CFG	3.5	39.4	2.0	**44.9**	**57.1**
Sonic-CFG	20.2	31.9	9.1	**61.2**	**47.9**
AutoSLM	29.9	12.4	7.1	**49.4**	**57.7**
AutoSLM + SRI SLM	23.7	8.2	8.6	**40.5**	**68.1**

Table 5: SemER results obtained for the transcriptions in Table 4.

	Collected Test User Utterance Set (20804 Utterances)						
	Error (%)						Total
	Mis Cat	In CFG	Out CFG	Ins	Del	Total	Correct
Oracle-SLM	1.3	2.9	3.1	1.2	0.2	**8.7**	**92.5**
Nuance-CFG	1.1	1.2	12.0	0.2	0.2	**14.7**	**85.6**
Sonic-CFG	13.1	3.0	13.6	2.1	0.4	**32.3**	**69.8**
AutoSLM	4.6	4.2	7.3	2.0	0.3	**18.4**	**83.6**
AutoSLM + SRI SLM	3.5	3.0	8.5	0.4	0.3	**15.7**	**84.7**

4. Experimental settings and results

To create a baseline result and to test our proposed SLM generation methodology, we collected a set of 20804 user utterances (live IVR application recordings) for 55 different prompts. A total of 23 CFGs/SLMs are needed to cover all the 55 prompts and on average, each prompt elicits responses with 10.09 different semantic categories. The baseline WER and SemER results for the 20804 utterance set are produced by the Nuance 8.5v commercial recognizer and SONIC [10], an ASR system from the University of Colorado at Boulder, using 23 CFGs (manually created and tuned by speech application designers). We use SONIC [10] again to test the SLMs generated by our proposed methodology. We trained the SONIC acoustic model for the telephone transcription task using 160 CallHome and 4826 Switchboard-1 conversation sides.

Table 4 presents the transcription WER results obtained for the various tests performed on our 20804 utterance test set. Table 5 presents the Semantic Error Rate (SemER) results obtained for the transcriptions in table 4. Each utterance transcription generated by the various systems presented in Table 4 is classified into one or more semantic categories using the semantic categorization technique presented in section 3. We used an absolute LCS threshold value of 65.0, a LCS difference threshold value of 5.0 and a

Table 6: Various possible semantic error scenarios for an utterance.

Transcription Semantic Category List Size	Reference Semantic Category List Size		
	> 1	$= 1$	$= 0$
> 1	MisCat, Ins or Del	MisCat, Ins	InCFG
$= 1$	MisCat, Del	MisCat	InCFG
$= 0$	OutCFG	OutCFG	

LCS difference decaying factor of 0.1. These values were derived by using the manual transcriptions of 20804 utterances as a development set and, we obtained a best SemER of 4.6%. We also need a ”NO-MATCH” category, which is assigned to the transcription when it does not map to any other category.

In Table 4, *MisCat* errors are due to mismatches between the category proposed by the transcription and the actual utterance category. *InCFG* errors are due to the transcription proposing a category while the utterance's actual category is a NO-MATCH. *OutCFG* errors are due to the transcription proposing a NO-MATCH while the utterance actually has a valid category. *Ins* errors are due to the insertion of a semantic category by the transcription while the utterance's actual category list does not contain such a semantic category. *Del* errors are due to the deletion of a semantic category present in the utterance's actual category list while the semantic category is missing in transcription's semantic category list. *Total Error (%)* is the sum of all the 5 different error counts divided by the total number of reference semantic categories. *Total Correct (%)* is $100 - MisCat(\%) - InCFG(\%) - OutCFG(\%) - Del(\%)$. Table 6 presents the various errors possible due to the variations in the number of categories proposed by the transcription and the number of categories present in the reference list.

The “Oracle-SLM” WER/SemER results are obtained by training the SLMs using the manual transcriptions for the 20804 utterances. The “Nuance-IVR” and “Sonic-CFG” WER/SemER results are obtained by the running Nuance and SONIC respectively, using 23 manually created CFGs. The “AutoSLM” WER/SemER results are obtained by running SONIC using the 23 SLMs automatically generated from our proposed methodology. The “AutoSLM + SRI HUB5 2000 SLM” WER/SemER results are obtained by running SONIC using the interpolation of the automatically generated SLMs with the SRI HUB5 2000 back-off trigram SLM. The SRI HUB5 2000 back-off trigram model is trained on Switchboard-1 (3M words), 100 Call-Home English conversations (0.21M words) and Broadcast News Hub-4 data (130M words). An interpolation weight of 0.9 is assigned to our SLMs while a weight of 0.1 is assigned to SRI HUB5 2000 SLM.

The “Oracle-SLM” obtains the best results for both WER (15.0%) and SemER (8.7%). These SLMs are trained on the manual transcriptions and therefore give the best possible results for obvious reasons. We obtain better baseline WER/SemER results using Nuance (44.9%/14.7%) with the manually created CFGs than with SONIC (61.2%/32.3%). In our opinion, this is due to the Nuance recognizer working better with CFGs than SONIC. The SLM automatically generated using our proposed system (“AutoSLM”) outperforms the “SONIC-CFG” WER/SemER results by 11.8% absolute WER reduction (19.6% relative reduction) and by 13.9% absolute SemER reduction (43.0% relative reduction). Our automatically created SLMs produce results which come very close to the performance of the manual CFG based Nuance recognizer. The parameters used to build “AutoSLM” were intuitively chosen and were not optimized for the test set using any empirical or statistical evidence. We chose the top 100 prefillers from the prefiller extraction module and combined these prefillers with all the synomyns found by the thesaurus module, with the lexical chain module using a threshold confidence score of 10. The interpolation of our SLMs with a general wide-domain SRI SLM produces the best result for WER outperforming both the “Nuance-CFG” and the “SONIC-CFG” results while getting closer to matching the performance of “Nuance-CFG” for SemER. We obtain a 4.4% absolute WER reduction (9.8% relative reduction) over the “Nuance-CFG” WER while the SemER is higher by 1%.

5. Conclusions

The deployment of IVR applications in a variety of domains makes it difficult to find any decent sized corpora to cover these domains. We presented a methodology to use different sources of knowledge easily available (spontaneous speech based prefiller patterns, WordNet and Roget's thesaurus and, WWW based statistical validation) and combine them together to produce domain specific SLMs which reduce the WER and minimize the SemER. The results show that our methodology can create good performing IVR SLMs while needing minimum amount of human intervention and prior knowledge regarding the expected user utterances.

6. Acknowledgements

We are grateful to Marta Tatu for many thoughtful suggestions and comments; Mark Hittinger and Cathie Ranta for their priceless encouragement and support; and to the anonymous reviewers for their valuable feedback.

7. References

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

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

[3] A. Acero, Y. Y. Wang, and K. Wang, “A semantically structured language model,” in *Proceedings of Special Workshop in Maui (SWIM)*, 2004.

[4] Q. Huang and S. Cox, “Automatic call-routing without transcriptions,” in *Proceedings of Eurospeech*, 2003.

[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 *Proceedings of AAAI Workshop on SLU*, 2005.

[6] E. K. Cave, M. Balakrishna, and D. Moldovan, “Efficient grammar generation and tuning for interactive voice response applications,” in *Proceedings of ICASSP*, 2006.

[7] D. Moldovan and A. Novischi, “Lexical chains for question answering,” in *Proceedings of COLING*, 2002.

[8] I. Bulyko, M. Ostendorf, and A. Stolcke, “Class-dependent interpolation for estimating language models from multiple text sources,” Tech. Rep., UWEETR-2003-0003, 2003.

[9] S. Schwarm, I. Bulyko, and M. Ostendorf, “Adaptive language modeling with varied sources to cover new vocabulary item,” *IEEE Trans. on Speech and Audio Processing*, 2004.

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