BOOTSTRAPPING LANGUAGE MODELS FOR SPOKEN DIALOG SYSTEMS FROM THE WORLD WIDE WEB

Dilek Hakkani-Tür

International Computer Science Institute
dilek@icsi.berkeley.edu

Mazin Rahim

AT&T Labs – Research
mazin@research.att.com

ABSTRACT

In this paper, we describe our approach for bootstrapping statistical language models for spoken dialog systems using in-domain web data and utterances collected from previous applications. The approach is based on the idea of stitching conversational templates with the predicate and arguments extracted from the web pages using semantic role labeling, to generate conversational style utterances. The conversational templates represent the task-independent portions of user utterances and can be built by hand, or learned from utterances collected from other domain applications. Experiments have shown that, stitching with both types of conversational templates have resulted in significantly better ASR word accuracy. Furthermore, the new language model bootstrapping approach can be combined with unsupervised and active learning to improve word accuracy even with very little in-domain transcribed data.

1. INTRODUCTION

This paper addresses the technical challenges behind the creation of statistical language models throughout the development life cycle of spoken language applications. In particular, we will focus on call routing applications that are supported by AT&T VoiceTone® [1]. VoiceTone® is a voice-enabled automated call center attendant that employs advances in spoken language technology to enable customers to access information and perform transactions by conversing naturally with a computer. The goal of these applications is not only to reduce operational cost of running call centers but also to improve customer experience over traditional Interactive Voice Response (IVR) systems.

One of the key challenges when creating VoiceTone applications is collecting a sufficient amount of in-domain data to train the statistical language models and semantic classification models. This process is not only resource intensive but also delays the time-to-deployment of the application. In this paper, we address the challenge of how to train language models for automatic speech recognition (ASR) by leveraging from the wealth of data from the World Wide Web. Although this may seem as a natural resource for creating language models, it is generally difficult to use since the statistics of the web language is vastly different than that observed in conversational style utterances. For example, the disfluencies, such as filled pauses or first/third person pronouns which are frequently observed in spoken language are rarely observed in the web data. Instead, there are often web-specific word sequences, such as "click on the link", which never occur in spoken dialogs. Nevertheless, there is sufficiently useful in-domain information, such as key phrases, product names, and abbreviations, that makes the web data a valuable resource for creating language models. In this paper, we describe a new method for generating conversational style utterances based on the idea of *stitching* conversational templates with in-domain predicate and arguments that are extracted from web pages. Our proposed method includes three steps; filtering, predicate/argument extraction, and stitching predicate and arguments to conversational templates to generate conversational utterances. The first step, filtering, removes the common task-independent sentences from the web text. Then, we semantically parse the web sentences, using the ASSERT semantic role labeling (SRL) tool [2] from the University of Colorado, and extract the predicate/arguments. The final step stitches the predicate and arguments into the corresponding slots of the conversational templates. The conversational templates can be manually written, or learned from a library of utterances collected from spoken dialog systems. We then merge the utterances generated using the web with the data collected from other applications, and build *n*-gram language models.

There have been several previous studies that have used data from the World Wide Web as an additional source of training data for language modeling. In [3], the marginal probabilities of a static *n*-gram language model are dynamically updated using the web data, to match the topic being dictated to the system. In [4], the *n*-gram counts estimated from the web are interpolated with traditional corpus-based estimates, resulting in a significant reduction in ASR word error rate. In [5], training data for creating a language model is supplemented with text from the web, and then it is filtered to match the style and/or topic of the target recognition task. In [6], we combined data from out-of-domain applications with in-domain web data to bootstrap language models, and used the resulting model for active and unsupervised learning [7].

This work was done while the first author was with AT&T Labs – Research.

1-4244-0469-X/06/$20.00 ©2006 IEEE

Recently, in [8] a language model is generated by combining external static text resources that are collected from other ASR tasks with dynamic text resources acquired from the web. The BLEU score [9] was used to select the relevant sentences. In [10], in-domain verbs and noun phrases are extracted from in-domain web pages and customer e-mails, collected from the domain by syntactic analysis. These are then used to artificially generate training data for the domain. In our approach, we use semantic parsing to extract relevant predicate and arguments from the web data for generating the initial training data for language modeling and then employ active and unsupervised learning. In addition, we examine the run-time performance in terms of accuracy versus run-time speed. This trade-off is important when deploying spoken dialog applications.

The organization of this paper is as follows. In the next section, we describe the major milestones for creating spoken dialog applications. In Section 3, we present our approach for stitching web data to bootstrap language models. We further extend this approach to include unsupervised and active learning, and present experimental results in Section 4. Section 5 provides a summary of this paper.

2. APPLICATION CREATION LIFE CYCLE

There are three milestones when creating a VoiceTone® application, or any spoken language application, that require building statistical language models for ASR; (a) Creation of an automated Wizard system [11], (b) deployment of the final service, and (c) refinement of the deployed service. For all these milestones, the challenge is to generate accurate and fast language models with as minimal in-domain data as possible. The accuracy is essential for overall service performance while the speed is critical to ensure scalability of the service.

Each milestone imposes a different technical challenge. The Wizard, which is a preliminary design of the application with the goal to collect conversational data, is designed and deployed without any in-domain speech data. The creation of the final service has the advantage of leveraging from the Wizard data collection but due to resource and timing limitations, the speech data is typically untranscribed. On the other hand, once the application is finally deployed, there is an abundant amount of data that is collected for service refinement - a process that typically spans several months. However, transcribing this data is expensive and time consuming, and hence, it is essential to rapidly identify a small subset of this data that when manually transcribed would provide the best trade-off in terms of system accuracy and speed.

In the next section, we will describe a unified framework for generating the data necessary in building or adapting statistical language models during the application creation life cycle, and propose methods for improving accuracy and speed of these models under the constraints imposed by the above three milestones.

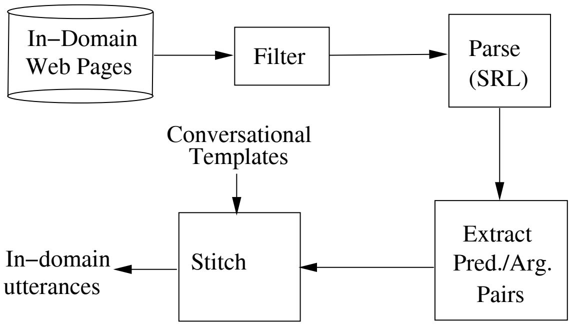

Fig. 1. The sequence of steps for generating in-domain utterances from the web data through stitching.

3. MODEL LEARNING

The new model learning comprises of multiple steps. We first learn domain-dependent conversational utterances using web pages related to the domain. We use these in combination with a library of utterances collected from previous applications, where applicable, to provide an initial language model for the automated Wizard system. Once data is collected from the Wizard, we use these with unsupervised learning to come up with improved language models for the domain. These models are used either during the Wizard phase or during the deployment of the service. The final phase is active learning, where based on available resources, we selectively transcribe domain-specific utterances to improve the deployed language models.

3.1. Learning through Stitching

Our approach for learning domain-dependent conversational utterances involves stitching conversational templates, extracted from spoken dialog systems, with predicate and arguments extracted from the in-domain web pages. Figure 1 shows the sequence of steps for generating in-domain conversational data. The first step is the filtering of the web data, W, so that the common task-independent web sentences (denoted by S), such as "Contact Us" or "Forgot your password?" are removed, forming the new set of sentences, $\hat{W}$:

$$\hat{W} = W - S. \quad (1)$$

The list of common web sentences can be obtained by taking the frequently occurring subset of sentences across multiple web sites.

The next step is extracting the predicate and arguments (the fillers) for the domain from the filtered web data. For this purpose, all the sentences are semantically parsed, and the predicates and arguments, PA, are extracted:

$$PA = find\_PA(sem\_parse(\hat{W})).$$

These are then used to generate in-domain conversational sentences, by stitching them to the conversational templates:

$$N = CT \,\hat{\circ}\, PA,$$

where CT is the set of conversational templates with their relative frequencies, and $\hat{o}$ is a stitching operation, which replaces the predicate and argument tokens in the conversational templates with the predicates and arguments from the domain while preserving the relative frequencies of the conversational templates.

Each conversational template is a sequence of words, with predicate and argument tokens. Some examples of conversational templates are:

uh I would like to [PRED] [ARG1],

I need to [PRED] [ARG1] please.

These templates can be either manually written, or learned using utterances from other applications. These out-of-domain utterances can be semantically parsed, and some of the predicates and arguments can be replaced by the predicate and argument tokens. For example, an utterance like:

[ARG0 I] would like to [PRED pay] [ARG1 my bill]

can be converted to the following template:

I would like to [PRED] [ARG1].

Note that the arguments, such as the [ARG0 I], which frequently have the same value are replaced with their values.

Once all utterances are processed as described, the templates frequently occurring in multiple applications data can be used as domain-independent conversational templates with their relative frequencies. In our experiments, we have tried using both manually written and automatically extracted conversational templates.

3.2. Unsupervised Learning

The goal of unsupervised learning is to use the untranscribed audio data from the application domain to generate improved language models. In this work, we use unsupervised learning with improved initial models learned through stitching, to generate models for the deployment of the final service. As we proposed in [7], we use the initial language model to recognize the audio data collected from the Wizard system. We then train a language model using the ASR output of these utterances. We showed that we get the optimum performance and speed when we only use the ASR output of in-domain utterances.

3.3. Active Learning

The goal of active learning is to select the smallest set of utterances that will have the biggest impact on performance when transcribed and added to the language model training [12]. In this work, we use active learning to further refine the language model used during deployment. We use the model resulted from unsupervised learning to recognize the audio files collected from the application, and select a subset of the utterances for transcription based on an utterance-based confidence score. We use the transcribed utterances to train an improved language model.

	N_1	N_2	$N_2 + L$
Vocab. Size	4K	4K	29K
OOV rate	8.7%	8.0%	0.7%

Table 1. The vocabulary sizes and OOV rates of various training sets. N_1 is the set learned through stitching via manual rules, N_2 is the set learned through stitching with automatically extracted rules, L is the library of out-of-domain utterances acquired from other applications.

Training Set	**Word Accuracy**
W	31.9%
N_1	41.9%
N_2	44.6%
L	67.9%
N_2 + L	68.7%
T (upper bound)	73.9%

Table 2. The ASR word accuracy results with various training sets. W is the web data, T is the in-domain transcribed training data.

4. EXPERIMENTAL RESULTS

4.1. The Data Set

The library consists of 462K out-of-domain utterances (3.4M words) taken from 8 spoken dialog applications. The vocabulary size of the library is 27.2K words. The in-domain training and test sets have 108K utterances (1M words) and 5.5K utterances (46K words), respectively. The purpose of the in-domain data is to run controlled experiments against the proposed techniques as will be shown later in this section. The vocabulary size of the in-domain training data is around 10K words and the percentage of out-of-vocabulary (OOV) words on the test set when using the library is 0.9%, and when using the in-domain training data is 0.6%. The in-domain web data contains 72K words, and the percentage of the OOV words on the test set when using the web data is 9.4%. This is significantly higher than the two other sets, and is a result of the fact that the web data does not include very frequent words in the conversational utterances such as the word *I*.

4.2. Learning Through Stitching

We used two sets of conversational templates to generate two sets of utterances, N_1 and N_2. N_1 was generated using manually built 5 templates, with equal frequencies. N_2 was generated using 81 templates learned from the library along with their frequencies. Table 1 shows the vocabulary sizes and the OOV rates on the test set, when using N_1, N_2 and a combination of N_2 and the library, L.

In order to test the performance of the language models trained using various data sets, we report word accuracy results on the test set in Table 2. When we use either N_1 or N_2, we get a significant improvement over using the web data, W, alone. When we merge N_2 with the library, L, we get a significant improvement over using only L [1]. We can achieve

[1] Using z-test with a confidence interval of 0.95.

Initial model training set	Word Accuracy
N_2	60.2%
L	71.4%
$N_2 + L$	71.8%

Table 3. Word accuracy with unsupervised learning using various initial models at run time of around $0.125 * \mathrm{realtime}$ (with a 2.4GHz CPU).

68.7% without any in-domain transcribed data, which is only 4.2% lower than the 73.9% that was obtained when transcribing all the in-domain training data of 108K utterances.

4.3. Unsupervised Learning

Once the Wizard is deployed with the language model trained using N_2 or $N_2 + L$ (when L is available), we are then able to collect utterances from the domain. Table 3 shows the word accuracy on the test data with unsupervised learning using various initial models. As can be seen, we can achieve 71.8% word accuracy at the operating point (selected just above the knee points in the run-time learning curve) without using any in-domain transcribed data, which is only 2.1% lower than the upper bound obtained using all the in-domain transcribed training data. At the operating point, we are able to support 8 simultaneous channels of ASR. Figure 2 shows the run-time learning curves with these language models. As can be seen from the figure, we obtain consistent improvements when we merge L with N_2.

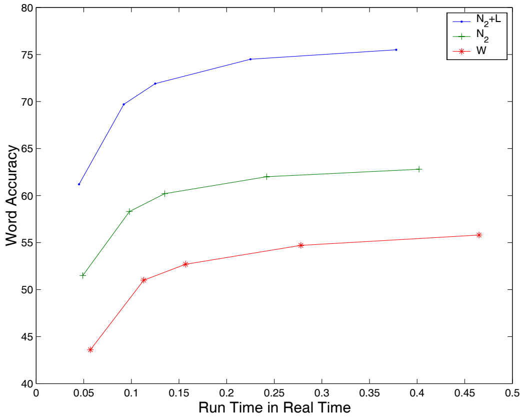

Fig. 2. Run Time in Real Time versus word accuracy curves after unsupervised learning with various initial models.

4.4. Active Learning

Once we collect data from the domain, and have resources for transcription, we can employ active learning. In this paper, we applied active learning using the 108K training utterances, and when we selectively sampled 11K utterances using active learning, we were able to match the upper bound 73.9% word accuracy. Therefore we are able to achieve the same accuracy using 108K in-domain utterances by only transcribing 10% of that data.

5. SUMMARY

We propose bootstrapping statistical language models for spoken dialog systems using in-domain web data and utterances collected from previous applications, and using these as initial models for unsupervised and active learning. Our bootstrapping approach is based on the idea of stitching conversational templates with the predicate and arguments extracted from the web pages using semantic role labeling, to generate conversational style utterances. We have shown that, stitching with both types of conversational templates have resulted in significantly better ASR word accuracy. Furthermore, when we combine stitching with unsupervised and active learning, we can achieve the same ASR word accuracy with 10 times less transcribed data.

Acknowledgements: We would like to thank Junlan Feng for providing us the web sentences.

6. REFERENCES

[1] M. Gilbert, J.G. Wilpon, B. Stern, and G. Di Fabbrizio, “Intelligent virtual agents for contact center automation,” *Signal Processing Magazine*, vol. 22, no. 5, pp. 32–41, 2005.

[2] S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky, “Shallow semantic parsing using support vector machines,” in *Proceedings of HLT/NAACL-2004*, Boston, MA, 2004.

[3] A. Berger and R. Miller, “Just-in-time language modeling,” in *Proceedings of ICASSP*, Seattle, WA, 1998.

[4] X. Zhu and R. Rosenfeld, “Improving trigram language modeling with the world wide web,” in *Proceedings of ICASSP*, Salt Lake City, Utah, 2001.

[5] I. Bulyko, M. Ostendorf, and A. Stolcke, “Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures,” in *Proceedings of HLT-NAACL*, Edmonton, Canada, 2003.

[6] G. Di Fabbrizio, G. Tur, and D. Hakkani-Tür, “Bootstrapping spoken dialog systems with data reuse,” in *In the Proceedings of SIGDIAL-2004, 5th SIGdial Workshop on Discourse and Dialogue*, Boston, MA, 2004.

[7] D. Hakkani-Tür, G. Tur, G. Riccardi, and M. Rahim, “Unsupervised and active learning in automatic speech recognition for call classification,” in *Proceedings of ICASSP*, Montreal, Canada, 2004.

[8] R. Sarikaya, A. Gravano, and Y. Gao, “Rapid language model development using external resources for new spoken dialog domains,” in *Proceedings of ICASSP*, Philadelphia, PA, 2005.

[9] K. Papineni, S. Roukos, T. Ward, and W. Zhu, “Bleu: A method for automatic evaluation of machine translation,” in *Proceedings of ACL*, Philadelphia, PA, 2002.

[10] M. Akbacak, Y. Gao, L. Gu, and H. J. Kuo, “Rapid transition to new spoken dialogue domains: Language model training using knowledge from previous domain applications and web text resources,” in *Proceedings of Interspeech*, Lisbon, Portugal, 2005.

[11] G. Di Fabbrizio, G. Tur, and D. Hakkani-Tür, “Automated wizard-of-oz for spoken dialogue systems,” in *Proceedings of Interspeech*, Lisbon, Portugal, 2005.

[12] D. Hakkani-Tür, G. Riccardi, and A. Gorin, “Active learning for automatic speech recognition,” in *Proceedings of ICASSP*, Orlando, FL, 2002.