RAPID LANGUAGE MODEL DEVELOPMENT USING EXTERNAL RESOURCES FOR NEW SPOKEN DIALOG DOMAINS

Ruhi Sarikaya[†], *Agustin Gravano*[‡] *and Yuqing Gao*[†]

† IBM T.J. Watson Research Center
Yorktown Heights, NY 10598
{sarikaya,yuqing}@us.ibm.com

‡ Columbia University
agus@cs.columbia.edu

ABSTRACT

This paper addresses a critical problem in deploying a spoken dialog system (SDS). One of the main bottlenecks of SDS deployment for a new domain is data sparseness in building a statistical language model. Our goal is to devise a method to efficiently build a reliable language model for a new SDS. We consider the worst yet quite common scenario where only a small amount (∼1.7K utterances) of domain specific data is available for the target domain. We present a new method that exploits external static text resources that are collected for other speech recognition tasks as well as dynamic text resources acquired from World Wide Web (WWW). We show that language models built using external resources can jointly be used with limited in–domain (baseline) language model to obtain significant improvements in speech recognition accuracy. Combining language models built using external resources with the in–domain language model provides over 20% reduction in WER over the baseline in–domain language model. Equivalently, we achieve almost the same level of performance by having ten times as much in–domain data (17K utterances).

1. INTRODUCTION

Language modeling (LM) is an important part of the speech recognition process. Much effort has been devoted to LM research. This has been concentrated in two directions [1]: 1) improving the language model probability estimation, 2) obtaining additional training material. Better probability estimation has been studied well [2]. However, methods of acquiring additional training data have received relatively little attention.

The largest data set available not only for language modeling but also for other natural language processing (NLP) tasks is the Word Wide Web (WWW), which currently consists of more than 4 billion pages[1], and is increasing at an astonishing rate. The Web has been utilized for a number of NLP applications [3]. We are aware of three recent studies in language modeling [1, 4, 5]. In [1], web–based n–gram counts are used to improve language modeling. The web–based counts are used to interpolate with the unreliable trigram estimates of the corpus based language model. Note that in this study the actual web pages are not downloaded, instead the n–gram counts returned by the search engine are used. In [4], an online language modeling paradigm is proposed where estimation and application of the language model are interleaved. The scheme uses a current language model to generate a speech recognition hypothesis for an utterance. The hypothesis is then sent to a web search engine, which returns relevant documents to update the corpus before rebuilding the language model either for the decoding of the next utterance or rescoring the current utterance. The queries are made up of content words without including any function words. Since the immediate context of the content words is not accounted for, such a search strategy typically retrieves text that is not conversational, hence may not be suited for SDS. In [5], domain independent conversational style data is retrieved from the web by using the most frequent trigrams in the Switchboard corpus as search queries. Pages with too many out–of–vocabulary (OOV) words were rejected.

There are two crucial issues, which are not given enough attention when using web data for language modeling: 1) query generation, 2) filtering the relevant text from the retrieved pages. We address the first issue by presenting a method to enlist queries from the most relevant to least relevant. The Web counts are certainly less sparse than the counts in a corpus of a fixed size. However, web counts are also likely to be significantly more noisy than counts obtained from a carefully cleaned and normalized corpus. So far, the decision to accept or reject a document had been usually based on very simple heuristics [3, 5]. It is thus inevitable that web–based counts contain a certain amount of noise. Unlike previous studies, we propose a principled mechanism based on a similarity measure for relevant data selection. The mechanism makes good use of limited in–domain data to sift through the large external text inventory to identify "similar" sentences. In many, if not all, of the previous work, web pages are used as the unit in accepting or rejecting training material. We believe that going one step further and sifting for relevant information is essential. Therefore, we do not take the returned documents as a whole, but rather find relevant utterances in the page. By doing so, we filter out irrelevant text and keep only relevant data for language modeling.

In practice when we start to build an SDS for a new domain, the amount of in–domain data for the target domain is usually small. In cases when there is no in–domain data, we generate artificial data. Using available data we build a pilot system that is mainly used to collect real in–domain conversational data. It is essential to have a pilot system that operates with a

[1] http://www.searchengineshowdown.com/features/google

	CORPUS	SIZE (Million Words)
Domain Independent	CTRAN	1.00
	Cellular	0.24
	FN-CMV	0.69
	Web-meetings	30
	SWB-Fisher	107
	UW web data	191
	Broadcast News	204
Domain Specific	Medical	1.4
	Call center1	0.33
	Call center2	1.9
	IBM Darpa Communicator	0.7

Table 1: Static data sources.

reasonable accuracy so that the users are not frustrated in communicating with the SDS. Moreover, if the system is not usable the training material will have unwanted artifacts. In deploying a successful pilot SDS having a reliable language model is essential. Unlike in acoustic modeling, language model quality depends heavily on the amount of in–domain data. In this study we attempt to address this issue by proposing a framework and a data selection technique to exploit external resources.

The rest of the paper is organized as follows. In Section 2 we present the proposed framework and focus on how to use external resources within this framework for language modeling. In Section 3, we explain how search queries are generated. The method for data selection from the retrieved text is discussed in Section 4. Section 5 presents the experimental results. Section 6 summarizes the findings and future research directions.

2. EXPLOITING OUT–OF–DOMAIN RESOURCES

We categorize external resources as static or dynamic. Corpora collected for other tasks are examples of static resources. The Web is a dynamic resource; its content is changing constantly. So far, the Web has been used mainly for domain independent speech recognition tasks. For example, it is used for AP newswire transcription [4], Switchboard and ICSI Meeting transcription [5], and the spoken document retrieval task [1]. To the best of our knowledge this study is the first successful attempt at exploiting web for speech recognition in a limited domain spoken dialog system.

In addition to web based data we also consider using data collected for domain specific as well as domain independent tasks. In Table 1, we enlist the static corpora used for the experiments in this paper. The domain independent sources are commonly used for large vocabulary speech recognition [8]. The domain specific sources are used for different SDS applications. In the table, "Call center1" refers to a telecommunication company's call center data. "Call center2" refers data from one of IBM's internal call centers where customers having trouble with their computers call IBM for assistance. Medical data is collected for a speech–to–speech translation project that involves dialogs between doctors and patients. None of the domain specific corpora are related to the target domain that involves financial transactions.

The approach we take is summarized in Fig. 1. We assume that we are given a limited amount of data belonging to the target domain. This data can also be generated manually, after one becomes familiar with the domain. We generate queries from these sentences and search the web. The retrieved documents

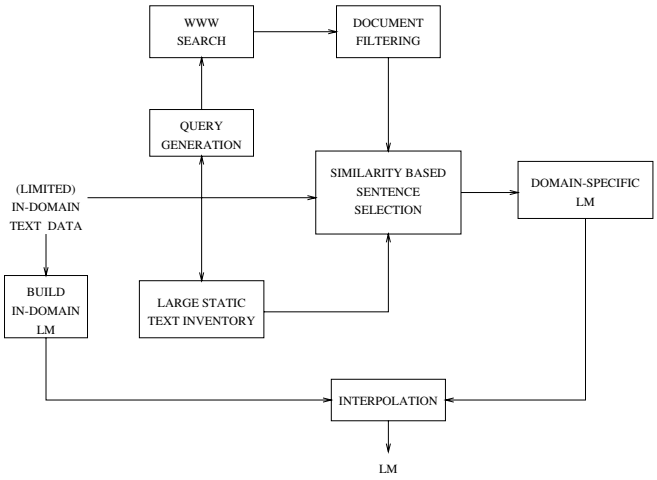

Figure 1: Flow diagram for the collecting relevant data.

are filtered. The documents are processed to extract relevant utterances using the limited in–domain data. We employ a similarity metric (see Section 4) to identify sentences that are likely to belong to the target domain. The same process is applied to the static data sources. However, in–domain data is directly used with the similarity metric to identify relevant utterances without query generation and retrieval. Finally, we build a domain–specific language model using the relevant sentences obtained from static and dynamic sources. An effective method to combine a small amount of in–domain and a large amount of out–of–domain data is through building separate language models and interpolating them [7].

3. GENERATING SEARCH QUERIES

As search engine we chose Google. Google indexes web pages (it also includes URLs that it has not fully indexed) and additional file types in the web database include PDF, .ps, .doc, .xls, .ppt, .rtf, .asp, .wpd, and more. Since, we did not want to deal with the conversion of these documents into plain text format and also due to their large sizes, we did not download these documents. We only downloaded those files where text can be retrieved efficiently.

There is a trade–off between the specificity of queries submitted to a search engine and the number of pages retrieved from the Web. Furthermore, the more specific a query is the more relevant the retrieved pages are. However, one of our concerns was to avoid too many repeated failed searches. Therefore, the approach we take for query generation is to start from the most relevant case and gracefully degrade to the least relevant case. We define the most relevant query as the one that has **AND**ed maximal n–grams with context. The least relevant query is defined as the **OR**ed unigrams obtained from an utterance. Web pages returned by Google for the most part contain (see example below) some of these n–grams. The objective in forming queries is to catch the topic and style.

As shown in Table 2, the first step in forming queries is to define a set of frequently occurring words as stop words (i.e. *the, a, is,*. etc.,). The remaining text is chunked into n–gram islands consisting of only content words. Then, we add context to these islands by including their left and right neighbors. Essentially,

(S)	what is the balance of my stock fund portfolio
	⇓ STOP-WORDS
	what *is the* balance *of my* stock fund portfolio
	⇓ N–GRAMS ISLANDS
	[what] [balance] [stock fund portfolio]
	⇓ ADD CONTEXT
(Q1)	[what is the] [the balance of] [my stock fund portfolio]
	⇓ RELAX N–GRAMS
	[what] [balance] [stock fund] [fund portfolio]
(Q2)	[what is the] [the balance of] [my stock fund] [fund portfolio]
(QN)	[what] [balance] [stock] [fund] [portfolio]

Table 2: Query generation.

we form n–grams having content words in the center. If a sentence starts with a content word, then we use the following two words as the context. Likewise, if a sentence ends with a content word, then we use the last two words before the content word as context. The goal in adding context around the content words is to incorporate conversational style into queries to some degree.

In Table 2, we present an example of query generation. We start with a sentence, "what is the balance of my stock fund portfolio". Next, we identify the stop words: *is, the, of, my.* The remaining word or phrase islands form the basis of the queries. Then, we add context to these islands. The amount of context can be increased by adding more neighboring words from the right and left of the content word. However, this will limit the number of hits and increase the number of failed requests from Google. We form queries starting with the most optimistic one (Q1), which combines n–gram chunks using **AND**. The next best query (Q2) is formed by splitting the trigram content word island, [stock fund portfolio] into two bigram islands, [stock fund] and [fund portfolio] and then adding context again. This is repeated until unigram islands are obtained. Note that the initial word islands can be as long as a sentence itself, if all the words are content words. In the example given above the largest island is a trigram. The queries [Q1, Q2,...,QN] are repeated by substituting **AND** with **OR** and added to the end of the query list. Note that in Google **AND** is implicit, therefore we did not insert **AND** between chunks when we form a query. During retrieval, queries from this list are submitted to Google until a pre–specified number of documents are retrieved. The retrieved documents are filtered by stripping off the HTML tags, punctuation marks and HTML specific information that is not part of the content of the page. The punctuation marks are used for sentence boundary detection.

4. SIMILARITY BASED SENTENCE SELECTION

Our objective is to find utterances in the external data sources that are semantically similar to a in–domain utterance. The key question is, what is the appropriate similarity measure? Inspired by the resemblance of the utterance selection problem to the machine translation (MT) evaluation problem where a translated or a candidate sentence is compared to a set of reference sentences, we adopted BLEU (BiLingual Evaluation Understudy) [6] as the similarity measure for utterance selection. BLEU is a fully automatic evaluation metric that forms a viable alternative to expensive and time–consuming human judgment of translation quality. Our confidence in the BLEU metric is reinforced by a statistical analysis of BLEU's correlation with human judgment for translation into English from four quite

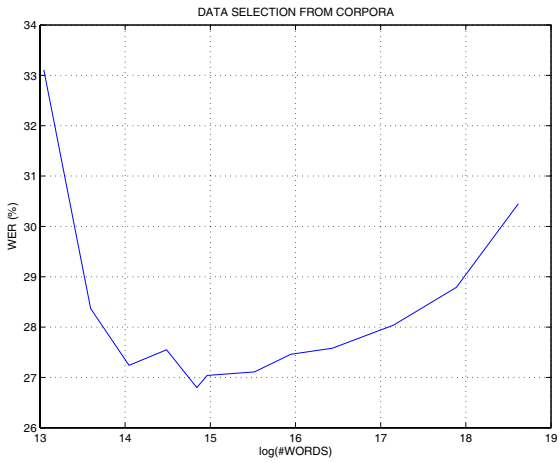

Figure 2: Data size versus WER using fixed external corpora.

different languages. The BLEU metric is defined as follows:

$$\text{BLEU} = \text{BP} \cdot \exp(\sum_{n=1}^{N} w_n \log p_n) \quad (1)$$

where N is the maximum n-gram length, w_n and p_n are the corresponding weight and precision, respectively, and BP is the brevity penalty:

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ \exp(1 - r/c) & \text{if } c \leq r \end{cases} \quad (2)$$

where r and c are the lengths of the reference and candidate sentences, respectively. The ranking behavior becomes more apparent in the log domain [6],

$$\log(\text{BLEU}) = \min(1 - \frac{r}{c}, 0) + \sum_{n=1}^{N} w_n \log p_n \quad (3)$$

Here, we used $N = 4$ and $w_n = 1/N$. We tailored the way BLEU is applied to our needs. Based on the analogy between MT evaluation and utterance selection, an in–domain sentence is treated as the "candidate" sentence and all the sentences in the external data containing at least one of the content words in the candidate sentence are considered as possible "reference" sentences. The task is to find a list of reference sentences with BLEU score greater than a pre–specified threshold. This threshold is determined based on word error rate (WER) using held–out data and is set to 0.08.

5. EXPERIMENTAL RESULTS AND DISCUSSION

The experiments are conducted on a financial transaction domain database. We assume that we have been given a reasonable vocabulary for the task along with the dialog states assigned to limited in–domain training data. The training data we use has 1.7K utterances uniformly selected from a larger set. The test data consists of 3148 utterances. The vocabulary has 3228 items. The acoustic models are trained using generic telephony data. All language models are dialog state based trigram with deleted interpolation. In all cases the data is split into a 90% and 10% chunks. The former chunk is used for training and the latter chunk is used as held-out set for smoothing.

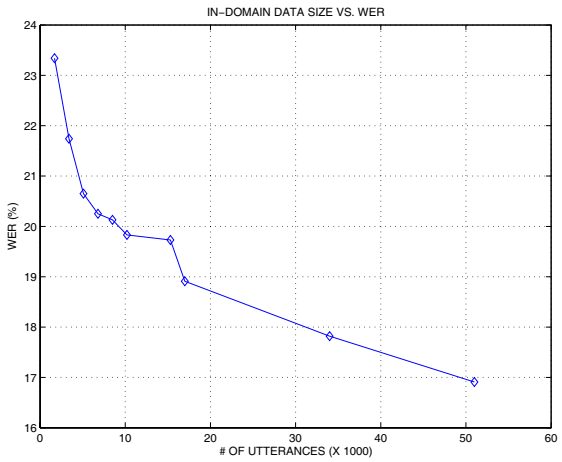

Figure 3: Data size versus WER using fixed external corpora.

Using a language model built for domain independent dictation or large vocabulary speech recognition tasks resulted in fairly high error rates ($> 45\%$). Therefore, using all the data we can possibly get without data selection or using domain independent language models for a limited domain task are not solutions. In Fig. 2, we plotted the WER with respect to the natural logarithm of amount of data retrieved from the static sources. Note that as the amount of data increases the relevance of the retrieved data decreases. The amount of data is increased by decreasing the similarity threshold. As we decrease the threshold the WER improves. However, beyond a certain point further reduction in threshold allows increasingly less relevant data to be used as part of the training data. Consequently, the WER starts to increase. The best performance is achieved by setting the similarity threshold to 0.08, which results in selection of 391K utterances (2.8M words).

For comparison, in Fig. 3, we plotted the WER against the amount of in–domain data. The first point in the graph is 24.3% corresponding to 1.7K utterances. As the data size increases the WER steadily improves. At 17K data size the WER is 18.9%. Note that this is a fairly low perplexity task. Even using only 1.7K sentences provides a fairly low WER.

Next, we combined the language models built using external resources with the in–domain language model as given in Table 3. The in–domain language model is built using a small amount of utterances (1.7K) and is taken as the baseline. Using only static corpora for language modeling (SCLM) resulted in 26.8% WER. We employed log–linear interpolation to combine the language models. Combining in–domain language model and SCLM reduced the WER to 21.7%. For dynamic data sources, we investigated setting the pre–defined limit to 20 pages/sentence vs. 100 pages/sentence for web data collection. Even if we set the pre–defined limit to 100 pages/sentence, the actual number of retrieved pages is on average around 60. This is due to disregarded file formats, inactive web sites and downloading problems. Using 20 pages/sentence (WWW–20) alone gave 24.1% (vs. 26.8% with the static corpora). Combining WWW–20 with the baseline language model achieved 20.2%. Increasing the number of retrieved pages to 100 (with no combination with the baseline) reduced the WER to 21.2% (WWW–100).

Performance of Language Models	
LM	WER (%)
Baseline (1.7K in–domain)	24.3
SCLM	26.8
Baseline + SCLM	21.7
WWW–20	24.1
Baseline + WWW–20	20.2
WWW–100	21.2
Baseline + WWW–100	19.2
Baseline + WWW–100 + SCLM	19.1

Table 3: Word Error Rates (WER) for various language model combinations.

Combining WWW–100 with the in–domain baseline language model resulted in 19.2%. Three way interpolation of SCLM, WWW–100 and in–domain language models resulted in the lowest WER: 19.1%. Overall, we achieved 5.2% absolute reduction in the WER compared to baseline. This figure is similar to what was obtained with 17K in–domain sentences, which gave 18.9%. It is interesting to note that web–based resources are more effective compared to static corpora.

6. CONCLUSIONS AND FUTURE WORK

We looked into ways to exploit static and dynamic text resources for building reliable statistical language models for spoken dialog systems. We presented methods for query generation and data retrieval from the World Wide Web. Furthermore, we introduced a new method based on a similarity measure for data selection from external resources. The method makes efficient use of the external text inventory minimizing the need for in–domain data. We showed a relative improvement of more than 20% in WER over the baseline in–domain language model. More importantly, we achieved virtually the same level of performance, if we had ten times more data. This is particularly important for building a pilot SDS system for collecting real data for the domain of interest. Next, we will concentrate on using syntactic and semantic information for data selection and will perform experiments using different amounts of in–domain data for financial as well as other domains.

References

[1] X. Zhu and R. Rosenfeld, "Improving trigram language modeling with the world wide web", *ICASSP-2001* pp. I:533–536, Salt Lake City, UT 2001.

[2] R. Rosenfeld, "Two decades of statistical language modeling: Where we go from here?", *Proceedings of IEEE*, vol. 88, no:8, 2001.

[3] M. Lapata and F. Keller, "The Web as a baseline: Evaluating the Performance of Unsupervised Web-based Models for a Range of NLP tasks", *HLT/NAACL*, pp. 121–128, Boston MA 2004.

[4] A. Berger and R. Miller, "Just–in–time language modeling", *ICASSP–98* pp. II:705–708, Seattle, WA 1998.

[5] I. Bulyko, M. Ostendorf and A. Stolcke, "Getting More Mileage from Web Text Sources for Conversational Speech Language Modeling using Class-Dependent Mixtures", *HLT-2003*, 2003.

[6] K. Papineni, S. Roukos, T. Ward and W. Zhu, "Bleu: A Method for Automatic Evaluation of Machine Translation", *Proc. ACL*, 2002, Philadelphia, PA.

[7] A. Rudnicky, "Language modeling with limited domain data", *Proc. ARPA Spoken Language Technology Workshop*, pp. 66–69, 1995.

[8] B. Kingsbury, *et al.,* "Toward domain-independent conversational speech reocgnition", *EUROSPEECH-2003*, Geneva, Switzerland 2003.