ICASSP 2010, Dallas TX

LANGUAGE MODEL ADAPTATION USING WWW DOCUMENTS OBTAINED BY UTTERANCE-BASED QUERIES

Andreas Tsiartas, Panayiotis Georgiou and Shrikanth Narayanan

Speech Analysis and Interpretation Laboratory
Department of Electrical Engineering,
University of Southern California,
Los Angeles, CA 90089

tsiartas@usc.edu, georgiou@sipi.usc.edu, shri@sipi.usc.edu

ABSTRACT

In this paper, we consider the estimation of topic specific Language Models (LM) by exploiting documents from the World Wide Web (WWW). We focus on the quality of the generated queries and propose a novel query generation method. In contrast to the n-gram based queries used in past works, our approach relies on utterances as queries candidates. The proposed approach does not rely on any language specific information other than the initial in-domain training text. We have conducted experiments with Web texts of size 0-150 million words, and we have shown that despite not using any language specific information, the proposed approach results in up to 1.1% absolute Word Error Rate (WER) improvement as compared to keyword-based approaches. The proposed approach reduces the WER by 6.3% absolute in our experiments, compared to an in-domain LM without considering any Web data.

Index Terms— Adapt language models, utterance queries, WWW corpora, in-domain documents

1. INTRODUCTION

An important source of information about Automatic Speech Recognition (ASR) is knowledge of how language is used. In ASR systems, language knowledge is represented by the Language Model (LM). LMs are often represented as a distribution of n-grams. Due to the general sparsity of domain matched spoken text corpora, researchers have investigated different approaches to estimate the prior probability of the n-grams not observed in the training text. Different approaches that have been proposed include manually providing additional utterances for training the LMs or using various back-off weight techniques, such as Kneser-Ney discounting. Recently, the Web has been used to estimate the probabilities of the unseen n-grams in the training text. The Internet provides a rich source of text documents spanning a large number of different topics and styles. In particular, forums, discussions on news articles, blogs, etc. provide good sources of conversational text that can potentially be used to train LMs. In this work, we use the Web to augment in-domain LMs, and we focus on the methods and quality of queries submitted to search engines.

The first work related to utilizing the Web for enhancing LMs was by Berger et al. [1] who used the hypothesis utterances of the ASR output as queries submitted to a search engine to mine topic specific documents from the Internet. In a similar fashion, Suzuki et al. [2] used the hypothesis transcript to identify nouns which are then used as queries to retrieve topic-specific documents. Other methods for extracting keywords from the ASR hypothesis transcripts include that proposed by Lecorve et al. [3]. In spite of these approaches being unsupervised, they heavily depend on the quality of the ASR hypothesis transcripts. Also, such approaches are not appealing for real-time systems since downloading documents and re-estimating the LMs can be computationally expensive and time consuming.

While the efforts described above assume that no prior topic information is available, various other methods have been proposed to generate queries from a training text using language specific information. For instance, in [4], Sarikaya et al. used a set of stop words to chunk the in-domain text into "n-gram islands". The "n-gram islands" are augmented by adding context and, finally, the queries are formed by combining the "islands" with AND and OR operations. In a different setup, Misu et al. [5] used a knowledge base to extract queries. In other works [6, 7], researchers have assumed the existence of some topic words. They extended a base vocabulary with selected topic words and used n-grams that include the topic words as queries. In spite of being a simple idea, this approach has the disadvantage that it might fail to represent topics already in the base vocabulary. In a multiple topics setup, Ng et al. [8] proposed a method for selecting topic-discriminant key-phrases to be used as queries from a set of 40 topics. Although having multiple topic scenarios is common in practice, training text that is already segmented into topics is not always available, as it was assumed in [8]. Availability of prior language information as assumed by [4, 5, 6, 7, 8] is also limiting and can be very costly when transitioning to new languages resulting in scalability issues for these existing methods.

Creutz et al. [9] proposed two multi-pass methods for generating queries. In the first approach, they incorporated extra linguistic information to select topic-specific queries. This approach has the same disadvantage as the methods described earlier. The second approach aims to form queries by selecting n-grams that are closer to an in-domain LM compared to the LM created from the first-pass of filtered in-domain Web text. Possible candidate n-grams are the top frequent n-grams from unigrams up to 5-grams. However, frequent n-grams have little content [7] and they are not very good for obtaining topically-matched data [7]. Also, in the first pass, they submit all the selected frequent n-grams to a search engine to gather statistics used in later passes. This can impose a limit on the number of n-grams considered, since query submission is an expensive operation. In addition, since the queries used depend on how close the n-grams are to the in-domain LM and how far they are from a Web in-domain LM, queries representing topics related to Web in-domain text might be ignored.

Futhermore, some other works assume that a topic-independent

language model exists. For instance, Wan et al. [10], proposed two methods for selecting topic specific n-grams by exploiting information from a background training corpus and an in-domain training text. Finally, Sethy et al. [11], used a background and an in-domain LM to extract keywords and key-phrases with good discriminative power using relative entropy. One of the disadvantages of the above approaches is that they require a background LM, which must be topic independent. Thus, the background LM must be chosen carefully such that it doesn't represent any topics of interest.

In this paper, we propose a novel approach to generate queries. Our queries are formed at the utterance level in contrast to past attempts that focused on the n-gram level. Moreover, our approach is language independent in the sense that it does not require any additional linguistic information about the language considered. Also, in contrast to [10, 11], our approach can be easily scaled to different languages and domains because it does not require any topic independent background LM. We show that the utterance-based queries can be beneficial in obtaining high quality in-domain documents and can outperform the previously proposed approach described in [11] that requires a topic-independent LM.

This paper is structured as follows. In the next section, we present a brief background of the query generation method. In section 3, we describe the system overview and how Web in-domain corpora are mined. In section 4, we explain the experimental setup and the evaluation methodology used in our approach. In section 5, we present the results of this work compared to prior efforts by work [11]. Finally, we summarize this work and propose some future directions.

2. SENTENCE BASED QUERY GENERATION

In contrast to past works that focus on queries generated by randomly combining n-grams, we work on queries that are complete utterances. Recently, search engines like Google have started to allow queries of length up to 32 words. This provides us with an opportunity to use much more context in each query. However, to be able to use such long queries, the terms need to be relevant and describe specific topics, otherwise it is unlikely the search engines will return any documents. For this reason, we chose to submit utterances as queries, which are more topic specific than combined n-gram queries. In addition, words in utterances are expected to be closely related.

2.1. Proposed query generation approach

In this section, we describe an iterative method to extract queries from a training set.

Let $\mathcal{T}$ be a training set of N unique utterances. Initially, we rank the utterances in $\mathcal{T}$ in descending order with respect to the number of words in each utterance. We pick the K utterances with the highest number of words. We denote by $\mathcal{Q}$, the set of K utterances selected from $\mathcal{T}$. Then, we update the set $\mathcal{T} = \mathcal{T} - \mathcal{Q}$.

In the first step, we randomly split the set $\mathcal{T}$ into two sets $\mathcal{T}_1$ and $\mathcal{T}_2$, with $|\mathcal{T}_1| = |\mathcal{T}_2| = \frac{|\mathcal{T}|}{2}$, if $|\mathcal{T}|$ is even [1]. One set is used to approximate the distribution of $\mathcal{T}$ and the other set to select the query utterances. Thus, we estimate the distribution of the utterances in $\mathcal{T}_1$, (i.e. the n-gram LM of the utterances in $\mathcal{T}_1$). We denote the probability of an event u in $\mathcal{T}_1$ as $\mathbb{P}(u|\mathcal{T}_1)$. Ideally, if $|\mathcal{T}_1|$ is large enough, it will approximate the distribution of the utterances in $\mathcal{T}$. Similarly, we estimate the distribution of the utterances in $\mathcal{T}_2$ and we denote the probability of an event u in $\mathcal{T}_2$ as $\mathbb{P}(u|\mathcal{T}_2)$. Likewise, we estimate the distribution of the utterances in $\mathcal{Q}$ and we denote the probability of an event u in $\mathcal{Q}$ as $\mathbb{P}(u|\mathcal{Q})$.

In the second step, we rank all utterances $u \in \mathcal{T}_2$ in descending order, according to the value given by $D(u, \mathcal{Q}, \mathcal{T}_1)$, which is defined as:

$$D(u, \mathcal{Q}, \mathcal{T}_1) = \frac{\log \mathbb{P}(u|\mathcal{T}_1)}{\log \mathbb{P}(u|\mathcal{Q})} \quad (1)$$

At this point, we pick the $\frac{K}{2}$ top utterances according to the above ranking and we denote the set of these utterances as $\mathcal{Q}_1$.

The third step is the same as the second step but the role of the sets $\mathcal{T}_1$ and $\mathcal{T}_2$ is switched. In this case, we rank all utterances $u \in \mathcal{T}_1$ in descending order, according to the value given by $D(u, \mathcal{Q}, \mathcal{T}_2)$. Then, we pick the $\frac{K}{2}$ top utterances according to the ranking value $D(u, \mathcal{Q}, \mathcal{T}_2)$ and we denote the set of these utterances as $\mathcal{Q}_2$.

Finally, we update the training set of utterances by $\mathcal{T} = \mathcal{T} - (\mathcal{Q}_1 \cup \mathcal{Q}_2)$ and the set of queries by $\mathcal{Q} = \mathcal{Q} \cup \mathcal{Q}_1 \cup \mathcal{Q}_2$. To obtain more queries, we use the updated sets $\mathcal{T}$ and $\mathcal{Q}$ and repeat the procedure, starting from the first step, until we get the desired number of queries.

2.2. Proposed approach properties

The approach described in section 2.1 starts with the K longest utterances as initial queries as they are the most likely to contain topic specific content. Then, we iteratively pick utterances that are close to the in-domain LM and far from utterances already considered as queries, according to eq. 1. At each step, the query set $\mathcal{Q}$ is augmented by adding the new queries and at the same time these utterances are removed from $\mathcal{T}$. Obviously, the sets $\mathcal{Q}$ and $\mathcal{T}$ must be disjoint at all steps.

An advantage of the proposed method is that it does not require any prior information about the language to generate the queries, apart from the initial in-domain training set. In addition, our approach tends to cover a broad range of in-domain topics since, at each iteration, we choose the utterances that are far from the already considered query utterances and close to the in-domain text.

Also, there is a trade-off when choosing the value of K. The value of K defines how many queries are extracted in each iteration. The distribution re-estimation is computationally expensive and, in each iteration, distributions are re-estimated. A high value of K leads to fewer iterations and, thus, faster execution times. On the other hand, more iterations result in more re-estimations of $\mathcal{Q}$ and $\mathcal{T}$ distributions, leading to more diverse queries covering a broader range of topics.

3. SYSTEM DESCRIPTION

In this section, we describe briefly the system used to build the in-domain LMs. A diagram of the system is shown in Fig. 1. Initially, we filter from the training text punctuations and non-alphanumeric symbols (we only keep the ' symbol). Then, the training text is passed to the query generator which picks the K available queries from the set of queries described in section 2. Afterwards, the obtained queries are passed to the query downloader which returns the URLs of the Web in-domain documents. Following that, the document URLs are passed on to the document downloader which returns the in-domain documents. At this point, we filter duplicate documents. The next step converts the html and the pdf documents into text and finally all documents are saved into UTF-8. Additionally, we estimate the utterance boundaries and we split the text into

[1] If $|\mathcal{T}|$ is odd, one of the two sets will have one more element than the other set.

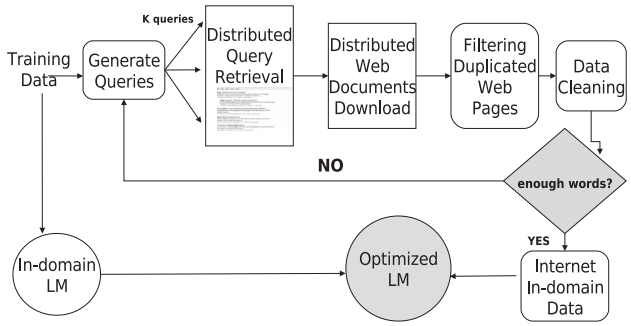

Fig. 1. This figure shows the general representation of the system used to mine, clean, and adapt Language models.

one utterance per line. Also, we filter the Web in-domain text from non-alphanumeric characters and punctuations (we only keep the ' symbol). The documents are merged and this procedure is repeated until we reach a prespecified number of words. Finally, the Web LM is merged with the in-domain LM using linear interpolation.

4. EXPERIMENTAL SETUP

In this section, we explain the experimental details of our work. The training set used is a subset of the DARPA Transtac English-Farsi collection corpus. The training set contains 32K utterances with doctor-patient interactions, car accident reports and interviews, directions, checkpoint interactions, etc. The search engine used is Google, which allows queries of size up to 32 words. Utterances longer than 32 words were not given to the query generator. In addition, we used only unique utterances to avoid repeated queries. After these steps our corpus was reduced to a total of 27K utterances, which we used as input to the system described in section 3. For each query, we retrieved the top 30 matching documents, if available. In this experiment we chose K=10 and estimated the distributions described in section 2.1 using 3-gram LMs. In addition, at each iteration, the query distribution is mixed with the in-domain distribution, weighting the in-domain distribution by a small weight (0.03). This ensures a common vocabulary of both the query and in-domain distribution.

Furthermore, we compared our approach with the query generator proposed in [11]. Because we focus on the quality of the queries, we didn't consider the cleaning part (i.e. selecting in-domain utterances) as described in [11]. Since this approach requires a specific number of keywords and key-phrases to be used, we optimized the parameters of this approach by downloading 5 million words of text and by experimenting with various combinations of keywords and key-phrases. Using perplexity as described in [11], we found that six keywords and two 2-gram key-phrases (6-2) minimized the perplexity. By using 3-gram key-phrases, we found that the perplexity was minimized when using two keywords, two 2-gram key-phrases, and one 3-gram key-phrase (2-2-1). Finally, by increasing the number of keywords and key-phrases, the search engine did not return enough documents and the perplexity was higher, so we did not consider those cases.

For both query generation methods, we downloaded Web text of size 5M, 10M, 25M, 50M, 100M and 150M words. We ran the ASR experiments and compared the quality of the queries using the Word Error Rate (WER) as an evaluation metric. For evaluation purposes we used Sphinx 3 trained on WSJ and TIMIT data with 12 MFCCs and energy along with first and second derivatives. The 3-gram language models generated for all conditions of our evaluation were generated using SRILM [12] using the Kneser-Ney discounting method. Finally, the Web in-domain LM is merged with the in-domain training LM using linear interpolation and optimized by minimizing the perplexity on a development text of 1600 utterances, as shown in Fig. 1. We used each merged LM to decode a set of 407 English utterances from a past DARPA evaluation of the system. The data is spontaneous and contains disfluencies.

5. DISCUSSION AND RESULTS

5.1. Discussion

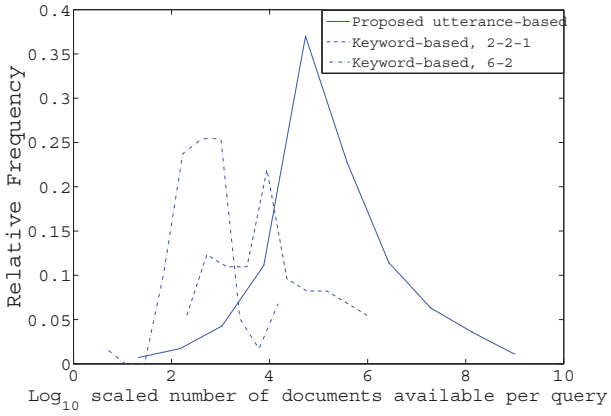

Fig. 2. The normalized histogram of the $\log_{10}$ scaled number of documents available per query. This histogram does not include samples when zero documents were returned

The end goal of the query-retrieval process is to obtain the required data but also the most relevant data; therefore the appropriate queries can significantly improve the efficiency and speed of the process [9]. Fig. 2 shows the normalized histograms of documents returned per query for the three query techniques under comparison. The horizontal axis denotes the $\log_{10}$ of the number of documents returned by the search engine per query. These samples were obtained by submitting 1000 queries to Google and by retrieving the top 30 documents. The proposed approach returns, in general, about 10 to 1000 times more documents per query than the keyword based one. Furthermore, some statistics reveal that our approach returns at least one document in 98.1% of the queries submitted compared to the keyword-based 2-2-1 and 6-2 which return at least one document in only 7.3% and 5.9% of the queries submitted, respectively. Google allows retrieving up to the top 1000 documents per query, even if more results are found. We found that our approach returns more than 1000 documents in 94.6% of the queries submitted compared to the keyword-based 2-2-1 and 6-2 which return more than 1000 documents in 4.9% and 1.4% of the queries submitted, respectively. The above-mentioned statistics show the efficiency of our proposed approach. Our method is efficient mainly because all terms used in a single query are related and describe one topic.

Fig. 3 shows the normalized histogram of the number of words per document in $\log_{10}$ scale. The figure shows that our proposed approach returns documents with fewer words than the keyword based approach. We believe the difference stems mainly from the use

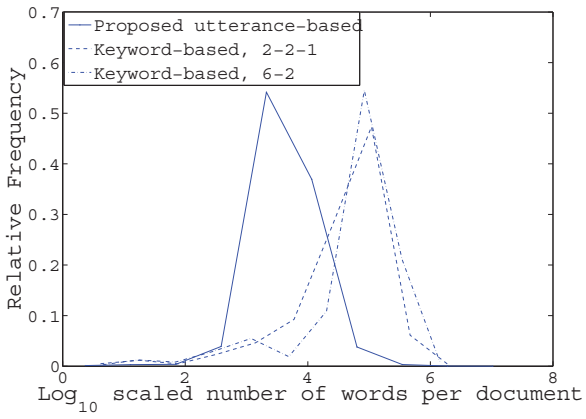

Fig. 3. The normalized histogram of the number of words per document in $\log_{10}$ scale.

of more topic-specific terms in the query that results in very domain specific documents. In contrast keyword-based querying returns much longer documents. The various terms and keywords are randomly combined for the query in those cases and often only very long documents can match all the terms. For example, these documents can be lengthy books or blog discussions with thousands of replies and, in general, might not represent a single topic.

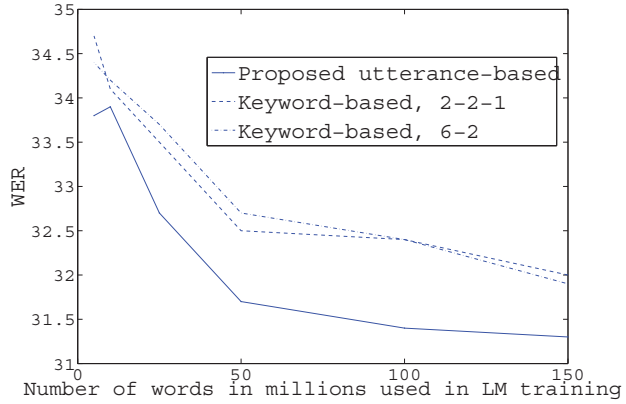

Fig. 4. WER against the number of words used to train the LM.

5.2. Results

The ASR performance as a function of the number of the words used in training the Web LM is shown in fig. 4. We can observe that the proposed utterance-based approach outperforms the n-gram-based approach proposed in [11] by up to 1.1% absolute reduction in WER. Considering only the in-domain LM, the WER is 37.6% and the proposed approach outperforms the only in-domain LM case by up to 6.3% absolute reduction in WER. It is notable that the performance gains of the proposed method increase as the Web-data used is increased. In particular, the difference is maximized when we consider 100 million words. The increase in the performance can be justified by the fact that the proposed approach covers a wider spread of topics as explained in section 2.2. In addition, the performance boost comes from the fact that utterance-based queries return documents that are closer to the domain of interest.

6. CONCLUSION

In this paper, we have introduced a novel method for generating high quality in-domain queries that do not require any language specific information except from an initial training set. In addition, we have conducted experiments with Web texts of size 0-150 million words and we have shown that the Word Error Rate (WER) is decreased by 1.1% absolute value using the proposed method, compared to the keyword-based work described in [11]. Also, the proposed approach reduces the WER by 6.3% compared to an in-domain Language Model (LM) without considering any Web data. For future work, we want to investigate different data cleaning methods and show performance improvement by selecting the documents and the utterances that are closer to the domain of interest.

7. REFERENCES

[1] Berger A. and Miller R., "Just-in-time language modelling," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Seattle, WA, USA, 1998, vol. II, pp. 705–708.

[2] Suzuki M., Kajiura Y., Ito A., and Makino S., "Unsupervised language model adaptation based on automatic text collection from www," in *Proceedings of Interspeech*, 2006, pp. 2202–2205.

[3] Lecorve G., Gravier G., and Sebillot P., "An unsupervised web-based topic language model adaptation method," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Las Vegas, NV, USA, 2008, pp. 5081–5084.

[4] Sarikaya R., Gravano A., and Yuqing G., "Rapid language model development using external resources for new spoken dialog domains," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005, pp. 573–576.

[5] Misu T. and Kawahara T., "A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts," in *Proc. ICSLP*, 2006, p. 912.

[6] Bulyko I., Ostendorf M., and Stolcke A., "Getting more mileage from web text sources for conversational speech language modeling using class-dependent mixtures," in *Proc. HLT-NAACL*, Edmonton, Alberta, Canada, 2003, p. 79.

[7] Bulyko I., Ostendorf M., Siu M., Ng T., Stolcke A., and Cetin O., "Web resources for language modeling in conversational speech recognition," *ACM Transactions on Speech and Language Processing*, vol. 5, pp. 125, December 2007.

[8] Ng T., Ostendorf M., Mei-Yuh Hwang, Manhung Siu, Bulyko I., and Xin Lei, "Web-data augmented language models for mandarin conversational speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2005, pp. 589– 592.

[9] Creutz M., Virpioja S., and Kovaleva A., "Web augmentation of language models for continuous speech recognition of sms text messages," in *Proceedings of the 12th Conference of the European Chapter of the ACL*, Athens, Greece, March 2009, pp. 157–165, Association for Computational Linguistics.

[10] Wan V. and T. Hain, "Strategies for language model web-data collection," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, Toulouse, France, 2006, vol. I, pp. 1069–1072.

[11] Sethy A., Georgiou P., and Narayanan S., "Building topic specific language models from webdata using competitive models," in *Proceedings of Interspeech*, Lisbon, Portugal, 2005, pp. 1293–1296.

[12] Stolcke A., "Srilm - an extensible language modeling toolkit," in *Proc. Intl. Conf. Spoken Language Processing*, Denver, Colorado, USA, September 2002.