

DATA DRIVEN APPROACH FOR LANGUAGE MODEL ADAPTATION USING STEPWISE RELATIVE ENTROPY MINIMIZATION

Abhinav Sethy, Shrikanth Narayanan

Speech Analysis and Interpretation Lab
Viterbi School of Engineering
Department of Electrical Engineering-Systems
University of Southern California

Bhuvana Ramabhadran

Human Language Technologies
IBM T. J. Watson Research Center
Yorktown Heights, NY

ABSTRACT

The ability to build domain and task specific language models from large generic text corpora is of considerable interest to the language modeling community. One of the key challenges is to identify the *relevant* text material in the collection. The text selection problem can be cast in a semi-supervised learning framework. Motivated by recent advancements in semi-supervised learning which emphasize the need of balanced label assignments, we present a stepwise relative entropy minimization scheme which focuses on selection of a set of sentences instead of selecting sentences solely on their individual merit. Our results on the IBM European Parliament Plenary Speech (EPPS) transcription system, show significant performance improvement (0.5% on an 8.9% baseline), with just a seventh of the out-of-domain data. The IBM EPPS LVCSR system which has a 60K vocabulary is a particularly hard baseline for out-of-domain adaptation because of low WER with in-domain training data.

Index Terms— Language model adaptation, speech recognition, relative entropy, TC-STAR, text mining

1. INTRODUCTION

An important step in creating speech recognition systems for different domains and applications is to identify the text resources for building the language models. In some cases text for the target domain might be available from institutions such as LDC and NIST. However in most cases the text is not readily available and needs to be collected manually. This imposes severe constraints in terms of both the system turnaround time and cost. To limit the effects of data sparsity a topic independent language model is often merged with a language model generated from limited in-domain data to generate a smoothed topic specific language model. However this approach can only be seen as a procedure to reduce the effect of data sparsity and will likely give suboptimal results to having good in-domain data.

This has naturally led to a growing interest in using the World Wide Web (WWW) as a corpus for building statistical models. Text harvested from the web combined with other large text collections such as GigaWord provides a good resource to supplement the in-domain data for a variety of applications. However, text gathered from such generic sources rarely fits the demands or the nature of the domain of interest completely. Even with the best queries and web crawling schemes, both the style and content of the data will usually differ significantly from the specific nature of the domain of interest.

For example, a speech recognition system requires conversational style text whereas most of the data on the web is literary.

The problem of extracting the relevant text from a generic collection can be seen as a semi-supervised [1, 2] learning problem. The dominant theme in recent literature on building language models with text acquired from the web is the use of various rank-and-select criteria for identifying sentences from the web-data¹ which match the in-domain data [3, 4, 5, 6, 7]. The central idea behind these schemes is to rank order sentences in terms of their match to the seed in-domain set, and then selecting the top sentences.

Ranking based selection is imbalanced and shows a natural bias towards selecting text material which has a high match with either the in-domain sentences or the in-domain model. Recent research in semi-supervised learning for classification ([2] presents a good survey) has shown the need to balance the unlabeled data. We believe that similar to the question of balance in semi-supervised learning for classification, we need to address the question of distributional similarity while selecting the appropriate sentences for building a language model from noisy data. Rank-and-select filtering schemes select individual sentences on the merit of their match to the in-domain model. To address the issue of distributional similarity we proposed a simple incremental selection algorithm which compared the distribution of the selected set and in-domain examples[8].

In this paper we present improvements to our text data selection algorithm and show its applicability on a state of the art LVCSR system used for transcription of European Parliamentary Plenary Speech (EPPS) [9] as part of the TC-STAR project. The TC-STAR (Technology and Corpora for Speech to Speech Translation) project financed by the European Commission within the Sixth framework Program is a long-term effort to advance research in speech to speech translation technologies². The primary goal of the TC-STAR project is to produce an end-to-end system in English and Spanish that accepts parliamentary speeches in one language, transcribes, translates and synthesizes them into another language, while significantly reducing the gap between the performance of a human (interpreter) and a machine. To support this goal, the performance of each component technology, namely, speech recognition (ASR), machine translation (MT) and text-to-speech (TTS) is optimized to produce the best output at their respective stages. The 2006 Evaluation was open to external participants as well as the TC-STAR partner sites [10].

Our experimental results on this system which was one of the best English LVCSR systems in TC-STAR evaluations, complement

¹We gratefully acknowledge support from NSF, DARPA, the U.S. Army and ONR

¹We will use web-data to refer to text harvested from web and other generic sources.

²Project No. FP6-506738

our previous results on the Transonics task [8]. The Transonics system is a real-time limited domain dialog system for medical domain conversations. For Transonics, the total available in-domain data set was limited to around 200K words. Sparsity of in-domain training material made it relatively easier to get improvements with out of domain data. The transcription system in contrast had good in-domain training material. The acoustic training transcripts provide 755K words and the final text editions of parliament speeches provide over 37M words for training the in-domain language model. Indeed, the availability of a good in-domain corpus has made it harder to get system improvements by using out-of-domain data [9].

The rest of the paper is organized as follows: The next section describes the data selection algorithm. Section 3 describes our strategy for gathering data from the web. Section 4 provides a concise description of the task and the baseline acoustic and language models. Experimental results and their analysis is presented in Section 5. We conclude with an overview of the paper and directions for future work.

2. BALANCED DATA SELECTION

The central idea behind text cleanup schemes proposed in recent language model adaptation literature for using web-data to build language models, has been to use a scoring function that measures the similarity of each observed sentence in the web-data to the in-domain set and assign an appropriate score. Various scoring mechanisms have been proposed in recent literature. In-domain model perplexity [3, 5] and variants involving comparison to a generic language model [7, 6] have been the dominant choice as ranking functions. A BLEU score based criterion to score out-of-domain sentences by computing their pairwise distance from individual sentences in the in-domain model was proposed by [4]. However ranking has an inherent bias towards the center of the in-domain distribution. This makes ranking a robust method for cleaning text but not for identifying small subsets which would be useful for building language models. Experimental as well as simulation results from [8] show very clearly the imbalance inherent in data selected by ranking.

Motivated by recent results in semi supervised learning [2] which show the importance of balanced selection we proposed an iterative selection algorithm [8]. The essential idea behind the algorithm, is to select a sentence if adding it to the already selected set of sentences reduces the relative entropy with respect to the in-domain data distribution. Based on experimental analysis of the performance of this selection algorithm, we came up with some critical modifications. In this section we present the new data selection algorithm and comment on how it compares with our basic scheme.

2.1. The Core Algorithm

Let us denote the language model built from in-domain data³ by P . Let $W(i)$ be the counts for words i in the vocabulary V of the P model. Our selection algorithm considers every sentence in the corpus sequentially. Suppose we are at the j^{th} sentence s_j . We denote the count of word i in s_j with m_{ij} . Let $n_j = \sum_i m_{ij}$ be the number of words in the sentence and $N = \sum_i W(i)$ be the total number of words already selected. The skew divergence of the maximum likelihood estimate of the language model of the selected sentences to the initial model P is given by

³The in-domain model P is usually represented by a linear interpolation of various text corpora available for the task

$$D(j) = \sum_i P(i) \ln \frac{P(i)}{(1 - \alpha)P(i) + \alpha W(i)/N}$$

The skew divergence [11] is a smoothed version of the Kullback-Leibler (KL) distance with the alpha parameter denoting the smoothing influence of the P model on our current Maximum Likelihood (ML) model. It is equivalent to the KL model for $\alpha = 1$. Using alpha skew divergence in place of distance was useful in improving the data selection especially in the initial iterations where the counts $W(i)$ are low and the ML estimate $W(i)/N$ changes rapidly. For notational simplicity, we denote $\beta = 1 - \alpha$. The model parameters and the divergence remain unchanged if sentence s_j is not selected. If we select s_j , the updated divergence is given by

$$D^+(j) = \sum_i P(i) \ln \frac{P(i)}{\beta P(i) + \alpha(W(i) + m_{ij})/(N + n_j)} \quad (1)$$

Direct computation of divergence using the above expressions for every sentence in the web-data will have a very high computational cost since $O(V)$ computations per sentence in the web-data are required. The number of sentences in the web-data can be very large, easily on the order 10^8 to 10^9 . The total computation cost for even moderate vocabularies (around 10^5) would be large.

However given the fact that m_{ij} is sparse, we can split the summation $D^+(j)$ into

$$\begin{aligned} D^+(j) &= \sum_i P(i) \ln P(i) + \\ &\quad - \sum_i P(i) \ln \left(\beta P(i) + \frac{\alpha(W(i) + m_{ij})}{N + n_j} \right) \\ &= D(j) + \underbrace{\ln \frac{(N + n_j)}{N}}_{T1} \\ &\quad - \underbrace{\sum_{i, m_{ij} \neq 0} P(i) \ln \frac{\beta P(i)(N + n_j) + \alpha(W(i) + m_{ij})}{\beta P(i)N + \alpha W(i)}}_{T2} \\ &\quad - \underbrace{\sum_{i, m_{ij} = 0} P(i) \ln \frac{\alpha W(i) + \beta P(i)(N + n_j)}{\alpha W(i) + \beta P(i)N}}_{\approx 0} \end{aligned} \quad (2)$$

Intuitively, the term $T1$ measures the decrease in probability mass because of the addition of n_j words to the corpus, and the term $T2$ measures the in-domain distribution P weighted increase in probability for words with non-zero m_{ij} . Using expression 2 makes it tractable to compute stepwise changes in divergence by reducing required computations to the number of words in sentence instead of the vocabulary size (Equation 1).

2.2. Selection and randomization

A sentence is selected if its inclusion decreases the divergence ($T2 > T1$). If a sentence is not selected we push it into a separate set of accumulated rejected sentences. We add the number of words in the sentence to an accumulation counter n_{rej} . We then consider the inclusion of the entire accumulated set into the set of sentences selected. $T1$ for the accumulated sentences can be calculated simply by using the above expression (substituting n with n_{rej}). To avoid calculation of $T2$ of the entire accumulated set everytime we add a

new sentence, we note that by Jensen's inequality T^2 for the accumulated set is upper bounded by the sum of individual T^2 for the rejected sentences. If this upper bound exceeds $T1$ we calculate the occurrence count m_i for every word in the rejected set and use that to calculate the T^2 for the accumulated set. In our previous version no accumulation was carried out.

The proposed algorithm is sequential and greedy in nature and can benefit from randomization of the order in which it scans the corpus. We generate random permutations of the sentence sequence and take union of the set of sentences selected in each permutation. Sentences that are included in more than two permutations are not considered for inclusion in other permutations, thus forcing the selection of different sets of sentences.

2.3. Further enhancements

To keep the description simple we have described the algorithm for the unigram case. It can be extended directly to higher order ngrams by considering tokens of size n as words. A more efficient implementation for back-off ngram models is to consider changes in back-off weight for every term seen in a sentence and propagating the relevant changes in counts $W_{ngram}(i)$ across higher back-off nodes. This however adds to the computational complexity since back-off computations cannot be marginalized in the same fashion we were able to marginalize the probability sums. Smoothing can be used after a fixed number of selected sentences to modify the counts of the selected text model $W(i)$. We have experimentally found out that Good-Turing smoothing after selection of every 500K words is sufficient for this task. The impact of smoothing was not seen to be significant to warrant further exploration.

A useful trick which boosts the performance of the algorithm is to reiterate through the selected set in reverse order. Order reversal is useful since initially $W(i)$ are low, which implies that the ratio $\frac{W(i)+m_{ij}}{W(i)}$ would be higher. This is also one of the motivations for moving to skew divergence instead of KL distance where the counts ratio is smoothed by the P model.

The next two sections describe the experimental setup. In the next section we describe our web-crawling scheme used to build a text corpus and in section 4 we describe the speech recognition system used as a baseline.

3. THE WEB CRAWLER

To generate queries for downloading relevant data from the web we use a technique similar to [3, 7]. An in-domain language model was generated using the training material and compared to a generic background model of English text [7] to identify the terms which would be useful for querying the web. For every term h in the language model we calculated the weighted ratio $p(h) \ln \frac{p(h)}{q(h)}$ where p is the in-domain model and q is the background model. The top scoring unigrams, bigrams and trigrams were selected as query terms. Starting from queries containing just trigrams we move to queries containing bigrams and then just unigrams. The set of URLs returned by Google are downloaded and non-text files are deleted. HTML files are converted to text by stripping off tags. The converted text typically does not have well defined sentence boundaries. We piped the text through a maximum entropy based sentence boundary detector to insert better sentence boundary marks. Sentences and documents with high OOV rates were rejected as noise to keep the converted text clean. We also computed the perplexity of the downloaded documents with the in-domain model and rejected text which

was above a threshold [7]. The initial size of the data downloaded from the web was around 750M words. After filtering and normalization the downloaded data amounted to 500M words.

4. ASR SYSTEM OVERVIEW

The 2006 IBM TC-STAR speech recognition system is organized around an architecture that combines multiple systems through cross-adaptation across different segmentation schemes and ROVER of the outputs from an ensemble of ASR systems. Training of acoustic models used EPPS material only. Each ASR system has approximately 6000 tied-states and 150K Gaussians. The acoustic front-end employs 40-dimensional, perceptual linear prediction (PLP) features obtained from an LDA projection that are mean and variance normalized on a per utterance basis. All systems employ Vocal Tract Length Normalization (VTLN), Speaker Adaptive Training (SAT) using features in a linearly transformed feature space resulting from applying fMLLR transforms, and are discriminatively trained on features obtained from a feature-space minimum phone error (fMPE) transformation (MPE models). A detailed description is provided in [9].

All decoding passes use a 4-gram modified Knesser-Ney model that was built using the SRI LM toolkit using the various sources described above. One model was trained on the training transcripts (LM1) and another on the text corpus based on the Final Text Editions (LM2). A perplexity minimizing mixing factor was computed using the Dev06 reference text. The final interpolated language model used in the construction of the static decoding graph contains 5.5M ngrams.

LM3 containing 80M ngrams was trained on 525M words of web data released by the University of Washington and LM4 containing 39M ngrams was built on 204M words of Broadcast News. The interpolation weights assigned to the out-of-domain language models LM3 and LM4 is relatively low, 0.12 and 0.13 compared to 0.21 and 0.54 for LM1 and LM2. The final interpolated LM contains 130M ngrams. The 59K recognition lexicon was obtained by taking all words occurring at least twice in the text corpus and once in the acoustic training transcripts. The OOV rate on the dev06 test set was slightly under 0.4%.

In the architecture described in [9], the best baseline system was obtained by rescoring the lattices produced after MLLR (speaker adaptation) with an out-of-domain language model (public condition). This is the only step that uses non-EPPS training material, i.e. UW web data and BN data.

The WER on the Dev06 and Eval06 system after LM rescoring with the out-of-domain LM, using a single system prior to ROVER was 11.0% and 8.9% respectively. The best performance after ROVER across multiple systems was 10.4% and 8.3%.

5. EXPERIMENTS

We present results on TC-STAR Dev06 and Eval06 test sets. The 2006 development set (Dev06) on which the acoustic and language models were optimized consists of approximately 3 hours of data from 42 speakers (mostly non-native speakers). The 2006 English Evaluation (Eval06) comprises 3 hours of data from 41 speakers. The Dev06 and Eval06 sets cover parliamentary sessions between June and Sept. 2005. Both Dev06 and Eval06 sets contain approximately 30K words. The text for Dev06 was used to fix LM weights for linear interpolation.

For the baseline system the in-domain language model was built with EPPS acoustic and final text transcriptions and was interpolated

Fraction of data selected(words)	All (500M)	1/11 (45M)	1/7 (71M)	1/3 (170M)
Perplexity(Dev)	94.5	94.5	91.3	88.7
Interpolation weight	0.32	0.29	0.45	0.49
WER (Eval)%	8.4	8.6	8.5	8.5
WER (Dev)%	10.7	10.9	10.8	10.6

Table 1. Performance comparison of the language models built with different fractions of data being selected for the Dev06 and Eval06 test sets. The baseline had 525M words of fisher web data (U.Wash) and 204M words of Broadcast News(BN) as out-of-domain data. The WER on Eval06 for the baseline was 8.9% and 11% on Dev06.

with out-of-domain LMs comprising U.Wash 525M word Fisher web corpus and 204M words from broadcast news (Section 4). The WER on Eval06 was 8.9% and 11% on Dev06. We provide performance comparisons against this baseline by replacing the two baseline out-of-domain LMs with LMs built from increasing fractions of text selected by our data selection method. As can be seen from Table(1), incorporating the 500M words mined by our crawling scheme boosted the system performance to 8.4% (6% relative) over the baseline. The effectiveness of the data selection scheme is demonstrated by the fact that we almost get the same WER gain (8.5 vs 8.4) and slightly better perplexity by using 1/7th of the data i.e 70M words. With 1/3rd data, we equal performance in WER terms and outperform significantly in perplexity. Combining the LM built from complete data with broadcast news decreased the WER to 10.6% and 8.3% on Dev06 and Eval06 respectively. In comparison, the LM built from 1/3rd data when interpolated with broadcast news gave 10.3% and 8.3% on Dev06 and Eval06 respectively, thus outperforming the LM built from entire data and BN.

5.1. Comparison to Transonics

It is interesting to compare the data selection results with those obtained for the Transonics [8] data set. For Transonics, we used a web corpus of 200M words. The data selection algorithm was able to achieve better performance than the out-of-domain LM built from the entire 200M word corpus, while selecting just 1/20th of the data. In contrast the IBM TC-STAR system requires a lot more data. However, if we consider the ratio of the selected data size with in-domain training data size we find the results much more comparable. This is expected since with good in-domain training data the dependency on out of domain data is less. In addition, the Transonic ASR system has a higher baseline WER than TC-STAR system.

5.2. Comparison with other TC-STAR component subsystems

More insights into these results can be gained by comparison with the performance of the ROVER-based TC-STAR system. Firstly, the 500M word web collection generated using the scheme presented here gave an improvement of 0.5% compared to the baseline which used two out-of-domain LMs (over 700M words). The data selection method is able to achieve the same improvement with just 70M words. Secondly, in the final stage of the TC-STAR decoding architecture, the baseline ASR output is combined with the output of three other systems using ROVER to achieve a reduction in WER from 8.9% to 8.3% and 11.0% to 10.4% on Eval06 and Dev06 test sets respectively. We are able to achieve close to that performance with just one system. Thirdly, the LM built from 1/3rd data interpolated with BN-based LM (LM3 in baseline) gives the same performance as the ROVER-based system.

6. CONCLUSION

We have presented a novel scheme for iterative data selection for building language models. This scheme not only results in significant gains to the best possible LVCSR system when compared to using a language model of comparable size built with no specific data selection, but also outperforms the best performing baseline (single) system. In fact, when interpolated with broadcast news data (a part of baseline LM) the performance improvements were equivalent to the gains observed by rovering three systems in the final version of the LVCSR system.

In the future we plan to extend the size of our out-of-domain corpus by mining more data from the web, and evaluate the effectiveness of data selection on larger sets for the EPPS task as well as its effect on bootstrapping language models for new tasks and domains.

7. REFERENCES

- [1] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell, "Text classification from labeled and unlabeled documents using em," in *Journal of Machine Learning*, 2000.
- [2] Xiaojin Zhu, "Semi-supervised learning literature survey," Tech. Rep. 1530, Computer Sciences, University of Wisconsin-Madison, 2005, http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.
- [3] Tim Ng, Mari Ostendorf, Mei-Yuh Hwang, Manhung Siu, Ivan Bulkyo, and Xin Lei, "Web-data augmented language model for mandarin speech recognition," in *Proceedings of ICASSP*, 2005.
- [4] Ruhi Sarikaya, Agustin Gravano, and Yuqing Gao, "Rapid language model development using external resources for new spoken dialog domains," 2005.
- [5] Teruhisa Misu and Tatsuya Kawahara, "A bootstrapping approach for developing language model of new spoken dialogue systems by selecting web texts," in *Proceedings of ICSLP*, 2006.
- [6] Karl Weilhammer, Matthew N Stuttle, and Steve Young, "Bootstrapping language models for dialogue systems," in *Proceedings of ICSLP*, 2006.
- [7] Abhinav Sethy, Panayiotis Georgiou, and Shrikanth Narayanan, "Building topic specific language models from web-data using competitive models," in *Proceedings of Eurospeech*, 2005.
- [8] Abhinav Sethy, Panayiotis G. Georgiou, and Shrikanth Narayanan, "Text data acquisition for domain-specific language models," in *Proceedings of EMNLP*, 2006.
- [9] B. Ramabhadran, O. Siohan, L. Mangu, G. Zweig, M. Westphal, H. Schulz, and A. Soneiro, "The IBM 2006 speech transcription system for european parliamentary speeches," in *Proceedings of ICSLP*, 2006.
- [10] "TC-STAR: Technology and corpora for speech to speech translation," <http://www.tc-star.org>.
- [11] Lillian Lee, "Measures of distributional similarity," in *37th Annual Meeting of the Association for Computational Linguistics*, 1999, pp. 25–32.