Text data acquisition for domain-specific language models

Abhinav Sethy, Panayiotis G. Georgiou, Shrikanth Narayanan
Speech Analysis and Interpretation Lab
Integrated Media Systems Center
Viterbi School of Engineering
Department of Electrical Engineering-Systems
University of Southern California

Abstract

The language modeling community is showing a growing interest in using large collections of text mined from the World Wide Web (WWW) to supplement sparse in-domain text resources. However, in most cases the style and content of the text harvested from these corpora differs significantly from the specific nature of these domains. In this paper we present a relative entropy (r.e.) based method to select *relevant subsets* of sentences whose distribution in an *n-gram* sense matches the domain of interest. Using simulations, we provide an analysis of how the proposed scheme outperforms filtering techniques proposed in recent language modeling literature on mining text from the web. A comparative study is presented using a text collection of over 800M words collected from the WWW. Experimental results show that by using the proposed subset selection scheme we can get performance improvement in both Word Error Rate (WER) and Perplexity (PPL) over the models built from the entire collection by using just 10% of the data. Improvements in data selection also translated to a significant reduction in the vocabulary size as well as the number of estimated parameters in the adapted language model.

1 Introduction

State of the art speech and Natural Language Processing (NLP) systems are data-driven, relying on learning patterns from large amounts of data. A key step in creating such systems for different domains and applications is to identify the appropriate text resources for building language models matched to that application or domain. In most cases, this data is not readily available and needs to be collected manually, which is an expensive and time consuming process.

This has naturally led to a growing interest in using the World Wide Web (WWW) as a corpus for building statistical models (Lapata, 2005; Lapata, 2002; Resnik, 2003). 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.

In most cases we have available to us a set of in-domain example sentences which we can use in a semi-supervised (Nigam, 2000; Zhu, 2005) fashion to refine our selection of text. Recent literature on building language models with text acquired from the web addresses the issue of mismatch, partly by using various rank-and-select schemes for identifying sentences from the web-data[1] which match the in-domain data (Ostendorf, 2005; Sarikaya, 2005). The central idea behind these schemes is to rank order sentences in terms of their match to the seed in-domain set, and then select the top sentences.

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

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 382–389,
Sydney, July 2006.

Research in semi-supervised learning for classification (Zhu (2005) 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. As a result, even though individual sentences might be good in-domain examples, the overall distribution of the selected set will focus solely on the high probability regions of the distribution. This will be made more clear in simulation results described in section 4.

To address the issue of distributional similarity we present an incremental selection algorithm which compares the distribution of the selected set and the in-domain examples by using a relative entropy (r.e.) criterion at each step. Some ranking schemes, which provide baseline for performance comparison are reviewed in section 2. The proposed algorithm is described in section 3. Section 4 will provide an analysis of how the proposed algorithm improves upon rank and select schemes. Experimental results are provided in section 5. We conclude with a summary of this work and directions for future research.

2 Rank and select methods for text cleaning

In recent literature, the central idea behind text cleanup schemes 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. The subsequent step is to set a threshold in terms of either the minimum score or the number of top scoring sentences. The threshold can usually be fixed using a held-out set. Ostendorf (2005) use perplexity from an in-domain n-gram language model as a scoring function. More recently, a modified version of the BLEU metric which measures sentence similarity in machine translation has been proposed by Sarikaya (2005) as a scoring function. Instead of explicit ranking and thresholding it is also possible to design a classifier to learn from positive and unlabeled examples (LPU) (Liu, 2003). In this system, a subset of the unlabeled set is selected as the negative or noise set **N**. A binary classifier is then trained using the in-domain set **I** and the negative set. The classifier is then used to label sentences in the web-data. The classifier can then be iteratively refined by using a better and larger subset of the **I/N** sentences selected in each iteration.

Rank ordering schemes do not address the issue of distributional similarity and select many sentences which already have a high probability in the in-domain text. Adapting models on such data has the tendency to skew the distribution even further towards the center. For example, in our doctor-patient interaction task, short sentences containing the word 'okay' such as 'okay','yes okay', 'okay okay' were very frequent in the in-domain data. Perplexity and other similarity measures assign a high score to all such examples in the web-data, increasing the probability of these words even further. In contrast other pertinent sentences seen rarely in the in-domain data such as 'Can you stand up please?' receive a low rank and are more likely to be rejected.

3 Incremental Selection

To address the shortcomings of the rank-and-select schemes we need to ensure that the set of sentences that we select from web-data has a distribution similar to the in-domain distribution. Thus we need to move from selecting sentences on the basis of individual score to selecting a set of sentences as a group. We propose an incremental greedy sentence selection algorithm based on relative entropy which selects a sentence if adding it to the already selected set of sentences reduces the relative entropy with respect to the in-domain data distribution. To describe our algorithm we will employ unigram probabilities though the method generalizes to higher order n-grams also.

3.1 The Basic Algorithm

Let us denote the language model built from in-domain data by P. We also need a language model P_{init} to initialize our selection algorithm . We experimented with two methods for selecting the initial model. The first is to use a uniform distribution over the vocabulary. The second approach is to sample with replacement the in-domain data and get a bagged estimate of the in-domain model. The results from both approaches were very close with the bagging approach being slightly better in all cases. For reasons of implementation simplic-

ity, we prefer the use of a uniform distribution. We convert the model P_{init} into a set of counts $W(i)$ for words i in the vocabulary V by multiplying with the vocabulary size V_n. Thus the total initial counts $\sum_i W(i) = V_n$.

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 relative entropy of the maximum likelihood estimate of the language model of the selected sentences to the initial model P is given by

$$H(j) = -\sum_i P(i) \ln \frac{P(i)}{W(i)/N}$$

The model parameters and the r.e. remain unchanged if sentence s_j is not selected. If we select s_j , the updated r.e. is given by

$$H^+(j) = -\sum_i P(i) \ln \frac{P(i)}{(W(i)+m_{ij})/(N+n_j)}$$

Direct computation of r.e. 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 $H^+(j)$ into

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

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}.

We will select the sentence s_j if including it decreases the r.e. with the in-domain distribution, i.e $H^+(j) < H(j)$. Thus s_j is selected if $T1 > T2$. To make the selection more refined we can impose a condition $T1 > T2 + thr(j)$ where $thr(j)$ is a function of j. A good choice for $thr(j)$ based on empirical studies is a function that declines at the same rate as the ratio $\ln \frac{(N+n_j)}{N} \approx n_j/N \approx 1/kj$ where k is the average number of words for every sentence. The counts W and N are updated with the selection of a sentence, and $H(j)$ is set to $H^+(j)$.

3.2 Text permutations and resequencing

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 permutations of the corpus by scanning through the corpus and randomly swapping sentences. Next we do sequential selection on each permutation and merge the selected sets.

As more sentences are selected, the r.e. $H(j)$ decreases and the distribution of the selected set gets closer to the in-domain distribution. The sentence selection becomes more refined as it becomes harder to improve $H(j)$ further. This motivated us to use a simple heuristic to ensure that the sentences selected in the initial stage are useful. At the end of a scan through the corpus we take the list of selected sentences and reverse it. We then scan the corpus again with the reversed list at the top. The sentences which were selected in the initial stages of the algorithm are retained only if they contribute to r.e. improvement even when they are in the latter part of the scan sequence.

3.3 Smoothing

The choice of maximum likelihood estimation for estimating the intermediate language models for $W(j)$ is motivated by a simplification in the entropy calculation which reduces the computation effort significantly. However, maximum likelihood estimation of language models is poor when compared to smoothing based estimation. To balance the computation cost and estimation accuracy, we modify the counts $W(j)$ using Good-Turing smoothing periodically, after a fixed number of sentences. The choice of the number of sentences to wait before smoothing depends on computation time constraints.

In the next section we provide an intuitive analysis of the advantages of iterative selec-

tion over rank-and-select schemes using simulated data.

4 Some intuition from simulations

A measure of the relevancy of the selected adaptation data is the Kullback-Leibler distance between the estimated n-gram model and the true n-gram distribution for that domain[2]. By comparing the two distributions we can identify the convergence properties of the data selection methods and also explain how the selection affects the estimated probability mass distribution compared to the true distribution.

To compare the n-gram language models we developed a fast r.e. computation scheme for tree based n-gram models. The description of this computation scheme is given in Appendix A. The fast scheme makes it possible to compute the r.e. between two LMs in $O(L)$ computations where L is the number of language model terms actually present in the two LMs, compared to V^n computations required in a direct implementation. This helps to reduce the computation effort by a factor of 10^4 or more.

We cannot use KL distance to judge relevance with real world data since the true distribution for real world data is unknown. However for analysis purposes, we can substitute the true distribution with a known distribution and then sample from it to get examples of in-domain text (with reference to the known distribution). We can then mix the known distribution with a noise model to simulate a generic text corpus, such as text acquired from web.

For simulation purposes, the language model used to generate the equivalent of in-domain text becomes the true distribution P_{true}. To complete the analogy with our data selection problem, text samples D_{ind} generated from P_{true} serve as the equivalent of in-domain data. The equivalent of the large generic corpus $D_{generic}$ can be generated from the noisy model. The simulation equivalent of the in-domain language model will be the language model P_{ind} estimated from the clean data D_{ind} .

We use a P_{true} with a vocabulary of 3K words estimated from a real world medical dialog task. The noise model is a language model of vocabulary 20K estimated from 1M words collected from webpages identified by Google using medical domain queries (Section 5.1). We used a D_{ind} set (generated from P_{true}) of 200K words and a generic set $D_{generic}$ of size 20M words or 3.8M sentences. The vocabulary sizes were kept small to efficiently generate text samples from the language models. The goal of the simulations is solely to gain an insight into the differences between iterative selection and rank-and-select scheme. Results on real world data are presented in the next section where the vocabulary size is more realistic.

[2]We restrict ourselves to the n-gram approach for modeling the distribution of word sequences.

	Rand	PPLSel	ItSel
200K	32.2	**9.1**	16.1
400K	34.2	**13.3**	24.3
800K	31.1	**22**	27.3
1200K	33.7	**28**	29.5
2400K	32.9	**31**	31

Table 1: Perplexity of data selected with respect to P_{ind} for varying number of selected sentences

4.1 Simulation results

The first question that we address is whether it is useful to select all data from the generic corpus $D_{generic}$ which scores high in perplexity terms with the in-domain model P_{ind}. In Table 1, we compare the perplexity of the data selected by different methods. *Rand* selects n sentences randomly, *PPLSel* selects the top n sentences ranked by perplexity and *ItSel* is the proposed iterative method. We build language models with the selected data and merge it with P_{ind} using weights determined from the heldout set. Table 2 shows the relative entropy of the adapted models with the true distribution. Our goal is to select the adaptation data cleverly to reduce the r.e. between the adapted model and the true distribution P_{true}. It can be seen that selecting data with lowest perplexity does not lead to a better language model. The perplexity of the data selected by *ItSel*, which is the most beneficial in improving the language model lies between the perplexity of random selection and *PPLSel*. It should be noted that by design, as the number of selected sentences approaches the size of the generic corpus, the selected data for all methods will be similar (identical if all the data is selected). Thus all methods essentially give same performance when selecting high percentages of data.

	Rand	PPLSel	ItSel
200K	12.1	15.2	**9.2**
400K	11.3	13.2	**8.3**
800K	10.5	11.1	**10.1**
1200K	9.7	9.5	**9.4**
2400K	9.3	8.9	**8.9**

Table 2: Relative entropy of models built from the selected data with the reference P_{true} distribution for varying number of sentences

Next we verify our hypothesis that use of ranking methods skews the distribution by focusing solely on the high probability regions. We selected the top 10% words which have the highest probability in P_{ind} and 10% words with the smallest probability. Then we computed the partial sums

$$H_{bias}^{high} = -\sum_{w \in top} P_{true}(w) \ln \frac{P_{true}(w)}{P_{sel}(w)}$$

$$H_{bias}^{low} = -\sum_{w \in bottom} P_{true}(w) \ln \frac{P_{true}(w)}{P_{sel}(w)}$$

for the language models P_{sel} estimated from the selected data. Note that the summation involves the true density P_{true}. If the selected data is imbalanced with respect to the true distribution P_{true} and unnecessarily biased towards the high probability regions of P_{base} the separation of the partial sums

$$H_{imbalance} = H_{bias}^{high} - H_{bias}^{low}$$

will be large. However, if the bias towards high probability regions of P_{base} is justified, $H_{imbalance}$ will be low.

In Figure 1, we plot $H_{imbalance}$ with increasing number of selected sentences for *PPlSel* and *ItSel*. High $H_{imbalance}$ for *PPLSel* especially when number of sentences selected is low confirms our hypothesis that selection using perplexity ranking skews the distribution by focusing solely on the high probability regions.

The simulation results provide an easy and intuitive way to understand how the proposed algorithm scores over rank-and-select scheme. Perplexity and WER results on real world tasks are hard to interpret because the underlying distributions are unknown. For example, our claim of skew towards high probability regions is hard to justify by looking at perplexity of test sets. However we do need to ensure that our algorithm indeed improves over the baseline methods on real world applications. We proceed to do so in the next section.

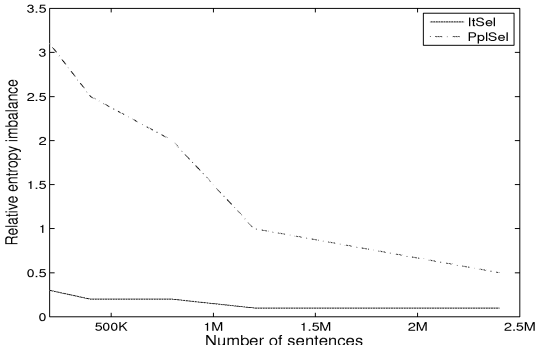

Figure 1: Relative entropy imbalance with number of selected sentences

5 Experiments

Our experiments were conducted on medical domain data collected as part of the English ASR of an English-Persian speech to speech translation project. We have 50K in-domain sentences for this task. A generic conversational-speech language model was built from the WSJ, Fisher and SWB corpora interpolated with the CMU LM. All language models built from web-data and in-domain data were interpolated with this language model with the interpolation weight determined on the heldout set. The test set for perplexity evaluations consists of 5000 sentences (35K words) and the heldout set had 2000 sentences (12K words). The test set for word error rate evaluation consists of 520 utterances.

5.1 Data collection from web

We downloaded around 100GB data from the web using automatically generated queries. Candidate query terms were generated by comparing the probabilities of word n-grams in the in-domain text with a background model of conversational speech. The prominent unigram, bigram and trigram word sequences were selected and combined to form queries for Google. The top 20 URLs returned by Google for each such conjunction of queries were downloaded and converted to text. The converted data from HTML 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

	10K	20K	40K
No Web	60.0	49.6	39.7
AllWeb	57.1	48.1	38.2
PPL	56.1	48.1	38.2
BLEU	56.3	48.2	38.3
LPU	56.3	48.2	38.3
Proposed	**54.8**	**46.8**	**38.1**

Table 3: Perplexity of testdata with the web adapted model for different number of initial sentences. Corpus size=150M

converted text clean. After filtering and normalization the downloaded data amounted to 320M words. We use a 150M word subset of this downloaded data for our initial experiments.

5.2 Experiments on 150M web-data

We first compare our proposed algorithm against baselines based on perplexity (PPL), BLEU and LPU classification (Section 2) in terms of test set perplexity. As the comparison shows, the proposed algorithm outperforms the rank and select schemes with just 10% of data. Table 3 shows the test set perplexity with different amounts of initial in-domain data. Table 4 shows the number of sentences selected for the best perplexity on the held-out set by the above schemes. *No Web* refers to the language model built from just in-domain data with no web-data. *AllWeb* refers to the case where the entire web-data was used.

The WER results are shown in Table 5. The average reduction in WER is close to 3% (relative). It can be seen that adding data from the web without proper filtering can actually harm the performance of the speech recognition system when the initial in-domain data size increases. This can be attributed to the large increase in vocabulary size which increases the acoustic decoder perplexity.

	10K	20K	40K
PPL	93	92	91
BLEU	91	90	89
LPU	90	88	87
Proposed	**12**	**11**	**12**

Table 4: Percentage of web-data selected for different number of initial sentences. Corpus size=150M

	10K	20K	40K
No Web	19.8	18.9	17.9
AllWeb	19.5	19.1	17.9
PPL	19.2	18.8	17.9
BLEU	19.3	18.8	17.9
LPU	19.2	18.8	17.8
Proposed	**18.3**	**18.2**	**17.3**

Table 5: Word Error Rate (WER) with web adapted models for different number of initial sentences. Corpus size=150M

5.3 Final results

To test how the performance of our algorithm scales with increasing data, we conducted experiments on a larger data set of 850M words which consisted of the medical domain collection of 320M words collected from the web and a 525M word collection published by the University of Washington for the Fisher corpus (Cetin, 2005). We will provide comparison with only the perplexity based rank-and-select system, as LPU and the BLEU based system are hard to scale to large text collections. Also, our results on the 150M set suggest that the performance of these systems is comparable to perplexity based selection.

The results on PPL and WER (Table 6, Table 7) follow the same trend as in the 150M data set. The importance of proper data selection is highlighted by the fact that there was little to no improvement in the unfiltered case (*AllWeb*) by adding the extra data, whereas there were consistent improvements when the proposed iterative selection algorithm was used. Perplexity reduction in relative terms was 7%,5% and 4% for the 10K,20K and 40K in-domain set, respectively. Corresponding WER improvements in relative terms were 6% ,4% and 4%. It is interesting to note that for our data selection scheme the perplexity improvments correlate surprisingly well with WER improvments. A plausible explanation is that the perplexity improvments are accompanied by a significant reduction in the number of language model parameters.

Table 8 shows the percentage of data selected using the proposed scheme and PPL based rank-and-select. We are able to achieve around a factor of 9 reduction in the selected data size. This translates to (Table 9) a factor of 7 reduction in the number of estimated language model parameters (bigram+trigram) and a 30% reduction in the vocabulary size.

	10K	20K	40K
No Web	60.0	49.6	39.7
AllWeb	56.9	47.7	38.2
PPL	55.8	47.4	38.2
Proposed	**52.6**	**45.3**	**37.1**

Table 6: Perplexity of testdata with the web adapted model for different number of initial sentences. Corpus size=850M

	10K	20K	40K
No Web	19.8	18.9	17.9
AllWeb	19.3	19.1	17.9
PPL	19.1	18.7	17.9
Proposed	**18.0**	**17.9**	**17.1**

Table 7: Word Error Rate (WER) with web adapted models for different number of initial sentences. Corpus size=850M

6 Conclusion

6.1 Contribution

In this paper we presented a novel and computationally efficient scheme for selecting *relevant* subsets of sentences from large collections of text acquired from the web. Our results indicate that with this scheme, we can identify significantly smaller sets of sentences such that the models built from the selected data have a substantially sparser representation and yet perform better (in terms of both perplexity and WER) than models built from the entire corpus. On our medical domain task we were able to achieve around 3% improvment in WER with a factor of 7 reduction in language model parameters while selecting a set of sentences 10% the size of the original web-data. The proposed method clearly outperforms text cleaning methods described in recent language modeling literature by moving from individual ranking to (greedy) group selection of sentences. Although our focus in this paper was on data acquired from the web, we believe the proposed method can be used for adaptation of domain specific models from other large generic corpora.

	10K	20K	40K
PPL	88.5%	87.8%	87.3%
Proposed	**9.5%**	**10.1%**	**8.9%**

Table 8: Percentage of web-data selected for different number of initial sentences. Corpus size=850M

	unigram	bigram	trigram
NoWeb	70K	1.5M	2.7M
AllWeb	105K	25.3M	36.2M
PPL	99K	22.1M	32.4M
Proposed	**70K**	**3.2M**	**8.2M**

Table 9: Number of estimated n-grams with web adapted models for different number of initial sentences for the case with 40K in-domain sentences. Corpus size=850M

We also present new analysis techniques (Section 4) based on simulated data and relative entropy which help us to gain valuable insight into the nature of different data selection algorithms.

6.2 Scope of this work

The research effort presented in this paper is directed towards selecting relevant domain specific data from large collections of generic text. We make no assumptions on how the data was collected or the use of specific web crawling and querying techniques. The methods we have developed can be seen to supplement the research effort by the machine translation community on identifying web resources (Resnik, 2003; Huang, 2005) or using web counts (Lapata, 2002) for language modeling.

6.3 Directions for future work

The proposed method can be combined with rank-and-select schemes described in Section 2. We are exploring the use of ranking to reorder the data such that the sequential selection process gives better results. Another idea we are currently investigating is to use multiple instances of the selection algorithm with different initial language models P_{init} generated by bagging. Sentence selection can then be improved by considering the composite entropy for all the models.

References

Bing Liu, Xiaoli Li, Yang Dai, Wee Sun Lee and Philip Yu. Building Text Classifiers Using Positive and Unlabeled Examples. Proceedings of ICDM. 2003.

Fei Huang, Ying Zhang and Stephan Vogel. Mining Key Phrase Translations from Web Corpora. Proceedings of EMNLP. 2005

Frank Keller, Maria Lapata and Olga Ourioupina. Us-

ing the Web to Overcome Data Sparseness. Proceedings of EMNLP. 2002.

Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun and Tom Mitchell. Text Classification from Labeled and Unlabeled Documents using EM. Journal of Machine Learning. 39(2:3)103–134. 2000.

Mirella Lapata and Frank Keller. Web-based models for natural language processing. ACM Transactions on Speech and Language Processing. 2(1),2005.

O.Cetin and Andreas Stolcke. Language Modeling in the ICSI-SRI Spring 2005 Meeting Speech Recognition Evaluation. ICSI Technical Report TR-05-006. 2005.

Philip Resnik and Noah A. Smith. The Web as a parallel corpus. Computational Linguistics. 29(3),2003.

Ruhi Sarikaya, Agustin Gravano and Yuqing Gao. Rapid Language Model Development Using External Resources For New Spoken Dialog Domains. Proceedings of ICASSP. 2005.

Tim Ng, Mari Ostendorf, Mei-Yuh Hwang, Manhung Siu, Ivan Bulyko and Xin Lei. Web-data Augmented Language Model for Mandarin Speech Recognition. Proceedings of ICASSP. 2005.

Xiaojin Zhu. Semi-Supervised Learning Literature Survey. Computer Science, University of Wisconsin-Madison.

R. C. Carrasco. Accurate computation of the relative entropy between stochastic regular grammars. RAIRO (Theoretical Informatics and Applications). 1997.

Appendix A: Fast Computation of Relative Entropy

We define the following symbols for the purpose of describing the r.e. computation:

x: The current word
h: The history $w_1..w_{n-1}$
h': The back off history $w_2..w_{n-1}$
b_h: The back-off weight for p distribution for history h
b'_h: The back-off weight for the q distribution
W: The vocabulary of the language model

Consider two n-gram language models $p(x|h)$ and $q(x|h)$.

r.e. at level n

$$D_n = \sum_{h \in H} p_h \sum_{x \in W} p(x|h) \ln \frac{p(x|h)}{q(x|h)} \quad (1)$$

We can divide the set of histories (H) at level n into H_s for all h which exist as $n-1$ gram and have a back-off weight$\neq 1$ in the p or the q distribution. The complement set ($H_{s'}$) will contain histories with a back-off 1. $H_{s'}$ corresponds to histories not seen in either language model. We define

$$D_h = \sum_{x \in W} p(x|h) \ln \frac{p(x|h)}{q(x|h)} \quad (2)$$

Then r.e. at level n D_n can be expressed as

$$\begin{aligned} D_n &= \sum_{h \in H_s} p_h D_h + \sum_{h \in H_{s'}} p_h D_h \\ &= \sum_{h \in H} p_h D_{h'} + \sum_{h \in H_s} p_h D_h - \sum_{h \in H_s} p_h D_{h'} \end{aligned}$$

Marginalizing w_1

$$D_n = D_{n-1} + \sum_{h \in H_s} p_h \left(D_h - D_{h'} \right) \quad (3)$$

D_h can be split into four terms depending on whether x/h is defined in the p or the q distribution

$$\begin{aligned} D_h &= T_1 + T_2 + T_3 + T_4 \\ T_1 &= \sum_{x \in X_1} p(x|h) \ln \frac{p(x|h)}{q(x|h)} \\ T_2 &= b_h \ln b_h \sum_{x \in X_2} p(x|h') \\ &\quad + b_h \sum_{x \in X_2} p(x|h') \ln \frac{p(x|h')}{q(x|h)} \\ T_3 &= \sum_{x \in X_3} p(x|h) \ln \frac{p(x|h)}{q(x|h')} - \ln b'_h \sum_{x \in X_3} p(x|h) \\ T_4 &= \sum_{x \in X_4} b_h p(x|h') \ln \frac{b_h p(x|h')}{b'_h q(x|h')} \\ &= b_h \ln \frac{b_h}{b'_h} \left(1 - \sum_{x \in X'_4} p(x|h') \right) + b_h D_{h'} \\ &\quad - b_h \sum_{x \in X'_4} p(x|h') \ln \frac{p(x|h')}{q(x|h')} \end{aligned} \quad (4)$$

Thus we are able to express D_h, in terms of the LM terms actually seen. Using D_h computed in this fashion in (3) we get a recursive formulation for r.e. at level n using LM densities actually seen. An alternative method for r.e. computation between finite state automata can be seen in (Carrasco, 1997).