IMPROVED TOPIC-DEPENDENT LANGUAGE MODELING USING INFORMATION RETRIEVAL TECHNIQUES

Milind Mahajan, Doug Beeferman[1], X. D. Huang

Microsoft Research
One Microsoft Way
Redmond, Washington 98052, USA

ABSTRACT

N-gram language models are frequently used by the speech recognition systems to constrain and guide the search. N-gram models use only the last N-1 words to predict the next word. Typical values of N that are used range from 2-4. N-gram language models thus lack the long-term context information. We show that the predictive power of the N-gram language models can be improved by using long-term context information about the topic of discussion. We use information retrieval techniques to generalize the available context information for topic-dependent language modeling. We demonstrate the effectiveness of this technique by performing experiments on the Wall Street Journal text corpus, which is a relatively difficult task for topic-dependent language modeling since the text is relatively homogeneous. The proposed method can reduce the perplexity of the baseline language model by 37%, indicating the predictive power of the topic-dependent language model.

1. INTRODUCTION

Speech recognition systems use language model to predict the words with associated probabilities given the history. In other words, if h represents the history i.e. word sequence up to present, language model is responsible for generating $P(w/h)$ for all words w in the vocabulary.

N-gram language models [6][7] are widely used by speech recognition systems, particularly in the large vocabulary speech transcription task [3][5][12]. N-gram language models make the simplifying assumption that the probability of the next word depends only upon the last N-1 words i.e. $P(w/h) \approx P(w/w_1 w_2 .. w_{N-1})$. Smoothed N-gram language models[4][8][9][13] are robust and can be automatically trained from a large text corpus. However, number of parameters in a N-gram language model increases dramatically with increasing N. This results in an increase in the size of the model, the data required to train it and the number of states the search algorithm must maintain while using the language model in search process. Therefore, values of N, which are typically, used range from 2-4. This results in the loss of long-term context information.

Topic of discussion is one of the important components of the long-term context information that often gets lost when using a short history window. Topic of discussion is a dynamic concept. It can change over time within the same document and new topics of discussion can also get created with new developments. Such topic or style information plays a critical role in improving the quality of the static N-gram language model. For example, the prediction of whether the word following the phrase “the operating” is “system” or “table” can be improved substantially by knowing whether the topic of discussion is related to computing or medicine.

There are several approaches, which help in incorporating long-term context information in the language model. Trigger language model in [17] uses trigger pairs derived using mutual information criterion. Trigger pair statistics is combined with N-gram statistics using Maximum Entropy framework. Trigger pairs provide long-distance information since the triggering and the triggered word can be separated by several words. Cache language models [11] boost the probability of a word or class seen in a long-term window over the history. Cache language models are similar to the trigger models with only self-triggers i.e. triggering and the triggered word in each trigger pair is identical. Cache models using the distance information to cause exponential decay in the probability of the same word being seen again are described in [2]. Long distance N-grams try to predict the next word based on N-1 words, which are not adjacent but are instead some distance back. [5] measured the information contained in distant bigrams. [16] combines distance-2 bigrams with standard bigrams and trigrams using maximum entropy. Tree-based language model in [1] is another approach to use longer context while limiting the number of parameters.

Domain or topic-clustered language models [10][15] split the language model training data according to topic. Division of the training data may be done using the known category information or using automatic clustering. In addition a given segment of the data may be assigned to multiple topics. A topic-dependent language model is then built from each cluster of the training data. Topic language models are then combined using adaptive linear

[1] Currently with School of Computer Science, Carnegie Mellon University, PA 15213

interpolation or other methods such as maximum entropy techniques.

Our approach to topic-dependent language models avoids any pre-defined clustering or segmentation of the training data. The reason for this is that the best clustering may only become apparent when the current topic of discussion is revealed. For example, when the topic of discussion is hand-injury to baseball player, the pre-segmented clusters of topic "baseball" & "hand-injuries" may have to be combined. This would lead to a union of the 2 clusters whereas the ideal dataset is obtained by the intersection of these clusters. In general, various combinations of topics can lead to a combinatorial explosion in the number of compound topics and it appears to be a difficult task to anticipate all the needed combinations beforehand.

We, therefore, base our determination of the most suitable language model data to build a model from, on the particular history of a given document. We use the known history of the document as a query against the entire language model training database of documents. Using well-known information retrieval techniques [14], we rank the documents in the database by relevance to the query. We then select the most relevant documents as the adaptation set for the topic-independent language model and adapt the topic-independent language model using this adaptation data. The process can be repeated as the document is updated. We demonstrate that the perplexity of the adaptive language model can be reduced by 37% in comparison to the baseline trigram language model, even for the relatively homogenous Wall Street Journal corpus.

The rest of this paper is organized as follows: In section 2, we outline the basic architecture used in dynamic language modeling, in section 3, we provide detailed information on the steps involved and on some variations on the basic algorithm of section 2. In section 4, we discuss the perplexity results of the experiments we conducted on the Wall Street Journal text corpus. Finally, in section 5, we discuss our conclusions.

2. DYNAMIC LANGUAGE MODEL ARCHITECTURE

There are two major steps in building the dynamic language model. The first step involves using the available document history to retrieve similar documents from the database. The second step consists of using the similar document set retrieved in the first step to adapt the general or topic-independent language model.

Available document history depends upon the design and the requirements of the recognition system. If the recognition system is designed for live mode application, where the recognition results must be presented to the user with a small delay, the available document history will be the partially user corrected history of the document thus far. On the other hand, in a recognition system designed for batch operation, the amount of time taken by the system to recognize speech is of little consequence to the user. In the batch mode therefore, a multi-pass recognition system can be used and the document history will be the recognizer transcript produced in the current pass.

2.1 Locating Similar Documents

We use the well-known information retrieval measure called tf-idf to locate similar documents in the training database to the history of the current document. Tf-idf measure is based upon vector space model of document representation. Each document and the query is represented by a vector. Each element of the vector corresponds to a word (or a term) in the vocabulary. So if V is the size of the vocabulary of terms then a vector $[w_{i1}, w_{i2}, ..., w_{iV}]$ represents document D_i. j^{th} element w_{ij} is derived from the term frequency (tf) of the jth term in the document D_i and the inverse document frequency (idf) of the term over the entire database of documents. Term frequency of the jth term in document D_i is the unigram count of the term in the document (denoted by tf_{ij}). Inverse document frequency of the jth term over a given database of documents is defined as $idf_j = \frac{Total\ Number\ of\ Documents}{Number\ of\ documents\ containing\ j^{th}\ term}$. w_{ij} is then defined as $w_{ij} = tf_{ij} * \log(idf_j)$.

Similarity between the two documents is defined to be the cosine of the angle between the corresponding vectors. Therefore,

$$Similarity(D_i, D_j) = \frac{\sum_{k=1}^{V} w_{ik} * w_{jk}}{\sqrt{\sum_{k=1}^{V} (w_{ik})^2 * \sum_{k=1}^{V} (w_{jk})^2}}.$$

All the documents in the language model training database are then ranked by the decreasing similarity between the document and the query. Query consists of the history of the current document.

2.2 Using Similar Documents

From the ranked list of similar documents, the most similar documents are selected as the adaptation set for the topic-adaptive language model. The selection criteria are based upon the number of documents to be selected or the value of the similarity measure. We then built a dynamic topic trigram language model from the set of selected similar documents and combined it with the topic-independent trigram language model and other language models, such as, cache language model and domain-specific language models, using adaptive linear interpolation.

Linear combination of language models can be represented as: $P_{combined}(w/h) = \sum_i \lambda_i P_i(w/h)$ where $\sum_i \lambda_i = 1$. The interpolation coefficients λ_i are obtained by using EM algorithm to maximize the likelihood of the history of the current document. Although, the document history has been used to generate one of

the language models, i.e. the dynamic topic language model, the history has been used indirectly and sufficiently generalized by using information retrieval that we did not feel it necessary to employ deleted interpolation. Particularly, since the process is quite time-consuming even without the added cost of deleted interpolation.

3. IMPROVEMENTS AND OBSERVATIONS

3.1 Multiple Dynamic Topic Language Models

While selecting the most similar documents for building the dynamic topic language model from the ranked list of similar documents, we face the question of how many of the similar documents to select. We want the dynamic topic language model to be as specific as possible to the topic at hand which indicates that we should select only a few most similar documents at the top of the list. At the same time, we want the dynamic topic language model to be as robust as possible. Selecting a larger number of similar documents increases robustness to the errors in the similarity ranking and also increases robustness of the estimation process since more data is available. Thus, there is a trade-off between specificity and the robustness of the dynamic topic language model. We worked around this trade-off by building multiple dynamic topic language models with overlapping and increasing number of similar documents and using all these dynamic language models during linear interpolation. In section 4, we present results using 4 dynamic topic language models built with top 50, 100, 200 and 400 similar documents. Another alternative would be to weight the data based upon the rank or similarity measure when building the language model.

3.2 Improving Information Retrieval

Since the information retrieval step is a key step in building the dynamic topic language model, any improvement in the performance of information retrieval is likely to improve the resulting language model. We, therefore, experimented with some methods to improve the information retrieval step.

3.2.1 Stemming

In information retrieval the technique of stemming is commonly used [14]. In stemming, various derivative forms of a word are converted to a root form of the word or stem. Root forms are then used as the terms that constitute the vocabulary for the purposes of information retrieval. The reason for this is the belief that the different derivatives of the root form do not change the meaning of the word substantially and the similarity measure based on word stems would be more effective by ignoring differences in derivative forms. Our experiments found that dynamic language models benefited from the use of word base forms during information retrieval.

3.2.2 Stopwords

Stopwords are common words that are omitted from the vocabulary for information retrieval. This is done since it is believed that the common words like “of”, “an”, “the” do not convey any information about the meaning of the document. On a small subset, our experiments did not show any improvement with the stopwords and we did not use stopwords in further experiments. This might be due to the fact that some of the stopwords might be somewhat useful for locating documents with the same style, which would be beneficial when building language models.

3.2.3 Query Expansion

Query expansion process [18] tries to improve the query through either local or global analysis. We attempted using a simple local query expansion where we added the contents of the top 5 documents retrieved by the query to the query and re-ran the query on the training database. On a small subset, our experiments did not show any significant improvement and considering the added expense, we did not use query expansion in further experiments.

4. EXPERIMENTAL RESULTS

We performed experiments on the Wall Street Journal text corpus used in [17] to test our ideas. This text corpus contains Wall Street Journal articles from the period of 1987 to 1989. We used the non-verbalized punctuation format and used the standard normalized form of the text data that is included in the corpus. The training set contains approximately 38 million words. We used a vocabulary of 60,000 words which was derived from the training set based on unigram counts. Our test set consisted of a subset of 100 articles containing approximately 55,000 words from the test set defined in [17]. Article boundaries for both the training and the test set are known and were used in our experiments.

The baseline language model was a trigram language model with bigram and trigram count cutoffs of 10. We used a uniform distance cache with a window size of 500 words. In addition, we used static trigram domain models, trained automatically by clustering the training data into a required number of clusters using unigram perplexity as the distance measure, as described in [10]. “Static 10” and “Static 20” entries in Table 1 refer to 10 and 20 trigram language models obtained by using this method.

In order to build the dynamic topic language models, for each article in the test set, we assumed that the first 100 words of the article are known. We then used this known history of the article as a query to perform information retrieval against the training set of articles and built 4 dynamic trigram topic language models using the top 50, 100, 200 and 400 most similar articles in the training set. “Dynamic 4” entry in Table 1 refers to this set of 4 language models. “Dynamic 1” entry in Table 1 refers to a single dynamic trigram topic language model built using the top 100 most similar documents (i.e. just the second of the 4 dynamic language models). We also used the known 100-word history to estimate the linear interpolation coefficients of the various language models using the EM algorithm. We then used the remaining words in the article in the perplexity calculations. Out of vocabulary words are excluded from the perplexity computations.

Table 1 shows the effect on perplexity of various model combinations. First column of the table contains the row number, the second contains the language model components used, the third column contains the perplexity number and the fourth column shows the percentage reduction in perplexity from the baseline.

Table 1: Perplexity of various language model combinations

Row	Language Models	PP	% Reduction
1	Baseline	167.5	-
2	(1) + Cache	142.5	14.9
3	(2) + Static 10	120.7	28.0
4	(2) + Static 20	120.4	28.1
5	(2) + Dynamic 1	113.4	32.3
6	(2) + Dynamic 4	108.5	35.2
7	(2) + Dynamic 4 + Stemming	107.2	36.0
8	(2) + Dynamic 4 + Static 10 + Stemming	104.5	37.6

Rows 3 and 4 show that increasing the number of static domain language models from 10 to 20 results in only a slight decrease in perplexity. This indicates that the effectiveness of static domain models in this experiment has reached a saturation point and cannot be improved by simply increasing the number of static domain models. Row 5 shows that a single dynamic topic language model is able to improve upon this and a combination of 4 dynamic topic language models gives a further improvement. Row 7 shows that use of stemming as described in section 3.2 improves the perplexity slightly. Row 8 shows that adding static domain models to dynamic models also improves the perplexity.

5. CONCLUSIONS

We have outlined a new way of building a customized language model for any document using information retrieval methods. We have shown that dynamic topic language models generated using this technique are effective in reducing perplexity by performing experiments. Our experiments have shown that the dynamic topic language model can be combined with other language models though linear interpolation to produce even further reduction in perplexity. The proposed method can be readily used for improving speech transcription applications or conversational applications, where dynamically constructed trigram language models for the correct domain often make a noticeable performance difference.

6. REFERENCE

[1] L. Bahl, P. Brown, P. De Souza, and R. Mercer, "A Tree-Based Statistical Language Model for Natural Language Speech Recognition," in IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 37, pp. 1,001-1,008, July 1989.

[2] P. Clarkson and A. Robinson, "Language Model Adaptation using Mixtures and an Exponentially Decaying Cache," in Proc. IEEE ICASSP-97, pp. 799-802, 1997.

[3] DARPA, "Proceedings of the Broadcast News Transcription and Understanding Workshop," Lansdowne, Virginia, Morgan Kaufmann Publishers, 1998.

[4] I. Good, "The Population Frequencies of Species and the Estimation of Population Parameters," Biometrika 40, pp. 237-264, Dec. 1953.

[5] X. Huang, F. Alleva, H. Hon, M. Hwang, K. Lee and R. Rosenfield, "The SPHINX-II Speech Recognition System: An Overview", Computer Speech and Langauge, vol. 2, pp. 137-148, 1993.

[6] F. Jelinek, "Self-Organized Language Modeling for Speech Recognition," in Readings in Speech Recognition, Morgan Kaufmann, 1989.

[7] F. Jelinek, "Up From Trigrams," in Proc. Eurospeech-91, pp. 1037-1040, 1991.

[8] S. Katz, "Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer", IEEE Trans. on Acoustics, Speech, and Signal Processing, vol. 35, pp. 400-401, 1987.

[9] R. Kneser and H. Ney, "Improved Backing-off for M-gram Language Modeling", in Proc. IEEE ICAASP-95, pp. 181-184, 1995.

[10] R. Kneser and J. Peters, "Semantic Clustering for Adaptive Language Modeling," in Proc. IEEE ICASSP-97, pp. 783-786, 1997.

[11] R. Kuhn and R. De Mori, "A Cache-based Natural Language Model for Speech Recognition," in IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 12, pp. 570-583, June 1990.

[12] K. Lee, H. Hon and R. Reddy, "An overview of SPHINX speech recognition system," IEEE Trans. Signal Processing, pp. 35-45, Apr. 1990.

[13] H. Ney, U. Essen and R. Kneser, "On the Estimation of Small Probabilities by Leaving-One-Out," in IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 17, pp. 1202-1212, Dec. 1995.

[14] G. Salton, "Developments in Automatic Text Retrieval," in Science, pp. 974-980, Aug. 1991.

[15] K. Seymore and R. Rosenfield, "Using Story Topics for Language Model Adaptation," in Proc. Eurospeech-97, 1997.

[16] M. Simons, H. Ney and S. Martin, "Distant Bigram Language Modeling using Maximum Entropy,", in Proc. IEEE ICASSP-97, pp. 787-790, 1997.

[17] R. Rosenfield, Adaptive Statistical Language Modeling: A Maximum Entropy Approach, Ph. D. Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, Apr. 1994.

[18] J. Xu and W. Croft, "Query Expansion using Local and Global Document Analysis," in Proc. ACM SIGIR-96, pp. 4-11, 1996.