TOPIC-BASED LANGUAGE MODELS USING EM

Daniel Gildea and Thomas Hofmann

University of California, Berkeley, and International Computer Science Institute
1947 Center Street, Berkeley, California
gildea,hofmann@icsi.berkeley.edu

ABSTRACT

In this paper, we propose a novel statistical language model to capture topic-related long-range dependencies. Topics are modeled in a latent variable framework in which we also derive an EM algorithm to perform a topic factor decomposition based on a segmented training corpus. The topic model is combined with a standard language model to be used for on-line word prediction. Perplexity results indicate an improvement over previously proposed topic models, which unfortunately has not translated into lower word error.

1. INTRODUCTION

The goal of statistical language models is to assign probabilities to sequences of words, and their most prominent application is in speech recognition, where language models provide prior probabilities that help in disambiguating acoustically similar utterances. By virtue of the chain rule it is sufficient to estimate the probability $P(w_i|h_i)$ of a word w_i conditioned on the history of preceding words $h_i \equiv w_1^{i-1}$. The main challenge in language modeling is to deal with the combinatorial growth in the number of possible histories, which implies a data sparseness problem and prevents a straightforward empirical estimation of the required conditional probabilities. A simple but commonly applied strategy is to make a (n-1)th order Markov approximation $P(w_i|h_i) \approx P(w_i|w_{i-n+1}^{i-1})$ which yields the class of n-gram language models, where typically $n = 3$ (trigrams).

While trigram models and variants thereof have proven hard to improve upon, they are unable to take advantage of long-range dependencies in natural language. Several more recent approaches attempt to overcome this limitation: Variable order models [15] adjust the length of the utilized contexts dynamically dependent on the available training data. Cache models [13, 3] increase the probability for words observed in the history, e.g. by some factor which decays exponentially with distance. Trigger models [16] are more general in that they allow arbitrary word trigger pairs to be incorporated into an exponential model. Grammar-based techniques [12, 2] exploit syntactic regularities to model long-range dependencies. Finally, in topic mixture models [11] a number of language models (e.g., n-grams) are trained on documents of various topics and are then combined at runtime.

Our approach is closely related to the latter class of topic mixtures in that the proposed model is based on a topic decomposition [9],

$$P(w|h) \quad = \quad \sum_t P(w|t)P(t|h)\,. \qquad (1)$$

Here t is a latent class variable that is supposed to refer to different topics; $P(w|t)$ are topic-specific word probabilities or *topic factors*, and $P(t|h)$ are mixing proportions that depend on the history h. For notational convenience all parameters are summarized in a vector θ. A graphical model representation that emphasizes the bottleneck principle of the topic variable is depicted in Figure 1.

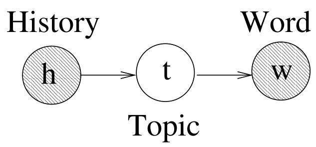

Figure 1: Graphical model representation of the topic factor model.

The main difference from clustering approaches like the one proposed in [11] is that we do not assume that each document or history belongs to exactly one topic cluster. Our approach is based on the less restrictive assumption of a low-dimensional approximation in terms of a linear combination of a small number of topic factors. A similar approach to language modeling based on a dimension reduction technique known as *Latent Semantic Analysis* (LSA) [7] has been proposed in [1] (a detailed implementation is provided in [4]). Yet, compared to the LSA approach, which makes use of Singular Value Decomposition techniques, our method has the crucial advantage of a strict probabilistic interpretation (cf. [9]), a fact that will be further discussed in Section 4.

The model we describe here does not make use of syntax and ignores the order in which words appear. In fact, implicit in (1) is the simplifying assumption that the influence of different topics on the statistical properties of language is limited to the level of single words (unigrams).[1] Local regularities can be taken into account at a subsequent stage, where we combine the topic model with a standard n-gram language model. We believe the model might also be profitably combined with a more sophisticated syntactic model such as those mentioned above.

2. TOPIC DECOMPOSITION BY EM

The "topics" used by our model are not taken from a predefined hand-labeled hierarchy, but rather emerge in a data-driven manner from the statistics of a corpus of training documents $d \in \mathcal{D}$. Based on the unigram assumption the data is reduced to simple word counts $n(w, d)$ of how often a word w was observed in a particular document d. All word counts can be summarized in the term-document matrix $\mathbf{N}$. As a training criterion we utilize the log-likelihood, i.e., the log-probability of the data under the model

$$l(\theta; \mathbf{N}) = \sum_w \sum_d n(w, d) \log \sum_t P(w|t)P(t|d)\,. \qquad (2)$$

[1] This is not a principled limitation of our model, yet it offers significant advantages in terms of computational complexity.

In the training procedure, the number of topics, i.e., the number of values the latent variable t can take, is predetermined, and the parameters $P(w|t)$ and $P(t|d)$ are fitted by the Expectation-Maximization (EM) algorithm [8]. Starting from randomly initialized values for the parameters this involves the standard procedure of alternating two computational steps: the E-step to calculate the posterior probability of the latent variables for given parameters and the M-step in which the parameters are re-estimated.

The E-step amounts to calculating the probability that a particular word w in a document d was generated by the topic factor t. For the rth iteration Bayes' rule yields

$$P^{(r)}(t|w,d) = \frac{P^{(r-1)}(w|t)P^{(r-1)}(t|d)}{\sum_{t'} P^{(r-1)}(w|t')P^{(r-1)}(t'|d)}. \quad (3)$$

The M-step adjusts the model parameters given the values for the latent variables calculated in the previous E-step:

$$P^{(r)}(w|t) = \frac{\sum_d n(w,d)P^{(r)}(t|w,d)}{\sum_{w'}\sum_d n(w',d)P^{(r)}(t|w',d)}, \quad (4)$$

$$P^{(r)}(t|d) = \frac{\sum_w n(w,d)P^{(r)}(t|w,d)}{\sum_{t'}\sum_w n(w,d)P^{(r)}(t'|w,d)}. \quad (5)$$

In our experiments, we used a modified (annealed) E-step (cf. [9]) to prevent overfitting. This amounts to introducing an exponent $0 < \beta \leq 1$ to discount the likelihood contribution in (3).

3. USING THE MODEL FOR TESTING

During testing, the $P(t|d)$ distributions computed for the training documents can be discarded, as they will not apply to new documents used for testing. Rather, we examine all the words seen so far in the document and calculate an estimate of $P(t|h)$ for the current history using only the topic factors $P(w|t)$. The mixing proportions $P(t|h)$ can be determined during testing by holding the probabilities $P(w|t)$ constant while estimating $P(t|h)$ by iterating (3) and (5) only over the words seen previously in the current document. Rather than doing the full EM calculation for $P(t|h_i)$ at each step during testing, we use an online approximation, calculated as follows:

$$P(t|h_i) = \frac{1}{i+1}\frac{P(w_i|t)P(t|h_{i-1})}{\sum_{t'}P(w_i|t')P(t'|h_{i-1})} + \frac{i}{i+1}P(t|h_{i-1}), (6)$$

$$P(t|h_1) = P(t) = \frac{\sum_{w,d} n(w,d)\,P(t|d)}{\sum_{w,d} n(w,d)}. \quad (7)$$

This is essentially an online EM algorithm of the type discussed in [14], but here only a single iteration is performed, reducing the computational complexity in the test stage to a minimum. Experiments using full EM iterations showed negligible improvements with higher computational costs.

Once the topic mixing proportions $P(t|h)$ have been determined, word probabilities can be calculated according to (1). As mentioned in the introduction, the topic model does not take advantage of short-range structure. Thus, we propose to combine the topic model with a standard language model which contributes a different type of information. For simplicity, we focus on combining it with an n-gram model. The combination scheme we favor is based on an intuition from maximum entropy model fitting by Iterated Proportional Scaling [6]. We interpret the topic model probabilities as marginal word distributions that should be preserved in the combined model, while leaving the higher-order structure unaffected. Under the assumption that the history h_i and the n-gram context are independent conditioned on w_i the following (approximation) formula can be derived

$$P(w_i|h_i, w_{i-n+1}^{i-1}) \approx \frac{P(w_i|h_i)P(w_i|w_{i-n+1}^{i-1})}{P(w_i)}. \quad (8)$$

Of course, this assumption is not valid in general as the $(n-1)$ word context of the n-gram model is part of the history, which implies that they are not even marginally independent. In our experiments, we have also evaluated two alternative interpolation methods of combining the n-gram and the topic-based model by averaging the respective probabilities (i) on the linear scale and (ii) on the log-scale. Both averaging schemes require an additional interpolation weight λ. Notice that the approximation by (8) as well as log-averaging require a re-normalization step.

4. EXPERIMENTS

4.1. Experimental Results on TDT-1

For our initial experiments, we used the TDT-1 corpus of newspaper text and transcribed broadcast news stories. The corpus contains 6,797,659 words in 15,862 documents. We formed our vocabulary by selecting all words occurring at least twice, which gave a vocabulary of 49,225 words. The data was augmented with sentence beginning and ending markers, but the symbols themselves were not counted in calculating perplexities. We used 90% of the data as a training set, holding out every 10th article for use in testing.

In a first series of experiments, we investigated the different schemes to combine the topic-based model with a conventional n-gram built from the same training data. The following table describes our test results on 4274 words comprising 10 stories. The number of factors in the topic model was 256, a restriction which was made due to complexity considerations. Although allowing a larger number of factors could in principle lead to overfitting, we found in practice that by using the control parameter β, the number of topics could be increased with no drop in test set performance.

Model	*Perplexity*
Unigram	1140.6
Topic model	829.1
Trigram	205.2
Linear interpolation: $\lambda P_{tri} + (1-\lambda)P_{topic}$	189.2
Log-scale interpolation: $\propto P_{tri}^{\lambda} * P_{topic}^{1-\lambda}$	180.8
Unigram rescaling: $\propto P_{tri} * \frac{P_{topic}}{P_{unigram}}$	170.1

Table 1: Results on the TDT-1 corpus

The rescaling method does not require an additional parameter fit and is nevertheless consistently superior than the interpolation schemes with optimized $\lambda = 0.9$ for linear and $\lambda = 0.8$ for log-scale interpolation. Using rescaling, a reduction of 17% in perplexity was achieved over the trigram model, which is relatively close to the 27% reduction from overall unigram to topic-based unigram perplexities.

To get a more reliable estimate of the model's perplexity, we ran the unigram scaling method over a larger test set of 24,850 words in 50 stories. Trigram perplexity was 180.8, whereas the combined model's perplexity was 147.2, a reduction of 18.6%. Perplexity reductions on individual stories ranged from 8% to 36%. Reductions on the five groups of ten stories ranged from 17.1% to 20.3%.

The improved perplexity results come with an increase in the computational load: normalizing over the vocabulary makes the computation of probabilities with the combined model slow, effectively increasing the complexity by a factor of the vocabulary size. Running on a 296MHz Ultrasparc roughly 10 words could be processed per minute, while the evaluation of a trigram model alone consists primarily of simple table lookups, and can process thousands of words per second. Experiments with a reduced 20,000 word vocabulary achieved a rate of 2 words per second.

4.2. Perplexity Results on the Wall Street Journal Corpus

In order to compare the performance of our probabilistic topic model with models based on standard LSA, we performed experiments using the same training/test data as in [4]. The training data consisted of 29,327,337 words in 81,553 articles taken from the Wall Street Journal from 1987, 1988 and 1989. The development test data consisted of 159,632 words from the same years, and the final test data consisted of 234,120 words from 1995 and 1996. We used the 19,979 word vocabulary provided with the corpus, augmented with sentence markers. Perplexity results are shown in Table 2.

	Dev-Test		Test	
Model	*Perpl.*	*Change*	*Perpl.*	*Change*
Unigram	1046.8		1107.7	
Topic	621.1	-41%	681.9	-38%
Bigram	174.3		235.5	
Bigram+Topic	134.5	-20%	187.3	-20%
Trigram	108.8		171.0	
Trigram+Topic	89.8	-17%	143.7	-16%

Table 2: Results on the Wall Street Journal corpus

The 20% improvement over bigram perplexity is significantly higher than the 12% improvement reported by [4] on the same data. This stresses the advantage of our probabilistic factor model that has also been verified in other applications [9, 10].

4.3. Analysis: When Does the Model Help?

It is interesting to consider which words the topic model helps in predicting. One might expect that because extremely common *function words* such as "and", "of", and "the" occur with approximately equal frequency in all documents, so that the topic model would be of little use in predicting them. In order to test this hypothesis, we calculated, for each vocabulary item in our test data, the average log ratio of the probabilities assigned by the two models:

$$\frac{1}{N}\sum_i log \frac{P_{topic+ngram}(w_i)}{P_{ngram}(w_i)}$$

One simple approximation of the distinction between *function* and *content* words is a word's overall frequency. We grouped vocabulary items by frequency to examine the correlation between the improvement yielded by the topic model and the word frequency. Results are shown in Figure 2. As can be seen, the topic-based language model actually performs less well than a trigram for roughly the 100 most frequent words in the data. Grouping words by their entropy over documents rather than frequency yields similar results.

These results suggest a simple modification to better handle function words: for function words use the n-gram probability directly, for other words combine the topic and n-gram probabilities as before, but now normalized only over the non-function words. The normalization constant is chosen such that the probability assigned to all non-function words by the combined model is the same as with the n-gram. Experiments showed that this approach did not in fact significantly lower the perplexity: 89.8 for the standard method vs. 89.4 for the function word method. However, the function word method has a beneficial side effect in terms of computational complexity, because it avoids the costly normalization when evaluating a function word.

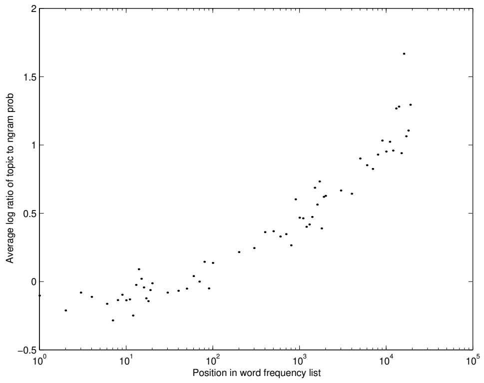

Figure 2: Relative performance of the topic model by word frequency.

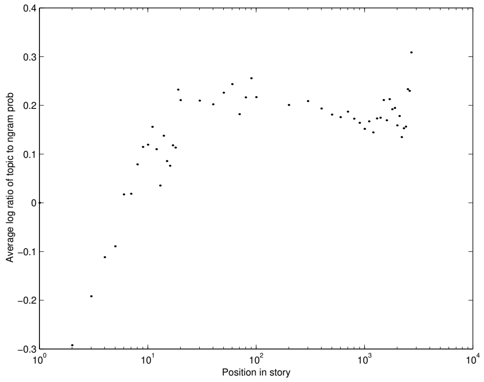

Figure 3: Relative performance of the topic model by position in story.

Another interesting way of analyzing the performance of the model is to look at how well it performs as a function of how many words of history are provided to the topic model. Such a graph is shown in Figure 3. As expected, the longer the history the more reliable the estimate of the article's topic, and the better the performance. The data also show that the combined model performs worse than the trigram for words 2 through 5 of a story. The gain from the topic model plateaus after the 19th word in the story.

5. APPLICATION TO SPEECH RECOGNITION: RESULTS ON BROADCAST NEWS

In order to determine how effective the topic-based language model is in a real-world application, we put it to use in a large vocabulary continuous speech recognition system. We used the SPRACH recognition system for broadcast news described in detail in [5]. For this experiment, we combined the trigram language model with a topic-based language model. The topic model used for the Broadcast News experiments was trained on 1996 CSR Hub-4 Language Model corpus, collected by the Linguistic Data Consortium. For efficiency, the 100 most frequent words were removed from the training data. After removing these words, the corpus training set consisted of 60,328,305 words spread over 124,814 documents. The trigram used had a vocabulary of 65,432 words; however for efficiency the topic model was trained on a vocabulary of only the 20,000 most common words, which cover 98% of the data.

Perplexity results were calculated both on test data from the CSR Hub-4 Language Model corpus and from two episodes of broadcast news for which acoustic data were available. These complete episodes were segmented into stories by hand. The trigram training data included the broadcast news transcripts used in training the topic model, as well as newswire text, for a total of roughly 450 million words. Perplexity results are shown in Table 3.

	Sprach 98 Trigram	Topic+ Trigram	# words in test set
LDC test data	155.6	134.3 (-14%)	76,260
CNN episode A	228.4	205.0 (-10%)	3412
CNN episode B	224.0	194.7 (-13%)	7554

Table 3: Perplexity results on Broadcast News

The percentage improvement over trigram perplexity is not as high as achieved with the WSJ corpus, probably because the trigram used in the WSJ experiments was trained on a relatively small amount of data. This result does show, however, that the topic model can still provide significant improvement over a state of the art trigram model. The improvement on the hand-segmented episode was smaller, no doubt due to the large number of extremely short "articles" in the data. Both of these shows contain many short headlines, summing up the news of the day, something not found in the Hub-4 training data.

Recognition results on CNN episode A actually deteriorated from 35.6% w.e.r. with the trigram model to 36.5% with the topic model combined with the trigram. One reason this topic-based model may not help as much as perplexity gains would indicate is that the model would tend to improve performance on longer content words, which are more easily acoustically distinguishable by the recognizer to begin with. In order to test this hypothesis, we categorized the recognizer's error according to the word's frequency.

	topic w.e.r	3gram w.e.r	# words
Frequent	27.4%	26.2%	1474
Rare	27.0%	27.4%	1816
Out of Vocab	73.8%	72.1%	122
Insertions (#)	276	244	

Table 4: Word error rate by word frequency

Table 4 shows results broken down into the 100 most frequent words (the words for which trigram probabilities are used), the remaining words in the topic model's vocabulary, words out of the topic model's vocabulary, and insertions. This analysis shows that the shorter frequent words are not in fact harder to recognize overall – the error rate is similar to rare words. However, the topic model does in fact give a small (not significant) improvement on the rare words.

6. CONCLUSIONS

Perplexity results on a variety of corpora show that topic-based EM techniques can be successfully applied to language modeling. The proposed technique performs better than a standard LSA-based method, and has a more intuitive probabilistic interpretation. Recognition results are disappointing, though a breakdown by word frequency gives some hope that the model may be able to help on content words. The fact that the topic model can also be applied in information retrieval [10] raises the prospect of using the same model for speech recognition and document indexing.

Acknowledgments Daniel Gildea was supported by a National Defense Science and Engineering Graduate Fellowship; Thomas Hofmann was supported by a DAAD postdoctoral fellowship.

7. REFERENCES

[1] J. Bellegarda. A latent semantic analysis framework for large-span language modeling. In *Eurospeech-97*, Rhodes, Greece, September 1997.

[2] C. Chelba and F. Jelinek. Exploiting syntactic structure for language modeling. In *COLING-ACL*, 1998.

[3] P. R. Clarkson and A. J. Robinson. Language model adaption using mixtures and an exponentially decaying cache. In *IEEE ICASSP-97*, pages 799–802, Munich, Germany, 1997.

[4] N. Coccaro and D. Jurafsky. Towards better integration of semantic predictors in statistical language modeling. In *ICSLP-98*, Sydney, Australia, November 1998.

[5] G. Cook, J. Christie, D. Ellis, E. Fosler-Lussier, Y. Gotoh, B. Kingsbury, N. Morgan, S. Renals, T. Robinson, and G. Williams. An overview of the SPRACH system for the transcription of broadcast news. In *DARPA Broadcast News Workshop*, Herndon, Virginia, February 1999.

[6] J. Darroch and D. Ratcliff. Generalized iterative scaling for log–linear models. *Ann. Math. Statist.* , 43:1470–1480, 1972.

[7] S. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantics analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1991.

[8] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

[9] T. Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the 15th Conference on Uncertainty in AI*, 1999.

[10] T. Hofmann. Probabilistic latent semantic indexing. In *22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA*, 1999.

[11] R. M. Iyer and M. Ostendorf. Modeling long distance dependencies in language: Topic mixtures versus dynamic cache model. *IEEE Transactions on Speech and Audio Processing*, 7, January 1999.

[12] D. Jurafsky, C. Wooters, G. Tajchman, J. Segal, A. Stolcke, E. Fosler, and N. Morgan. Using a stochastic context-free grammar as a language model for speech recognition. In *IEEE ICASSP-95*, pages 189–192, 1995.

[13] R. Kuhn and R. de Mori. A cache based natural language model for speech recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14:570–583, 1992.

[14] R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers, 1998.

[15] F. C. Pereira, Y. Singer, and N. Tishby. Beyond word n-grams. *Computational Linguistics*, June 1996.

[16] R. Rosenfeld. A maximum entropy approach to adaptive statistical language modeling. *Computer, Speech and Language*, 10, 1996.