A PLSA-based Language Model for Conversational Telephone Speech

David Mrva and Philip C. Woodland

Cambridge University Engineering Department
Trumpington Street, Cambridge, CB2 1PZ, UK
{dm312,pcw}@eng.cam.ac.uk

Abstract

This paper describes experiments with a PLSA-based language model for conversational telephone speech. This model uses a long-range history and exploits topic information in the test text to adjust probabilities of test words. The PLSA-based model was found to lower test set perplexity over a traditional word+class-based 4-gram by 13% (optimistic estimate using a reference transcript as history) or by 6% (realistic estimate using recognised transcript as history). Moreover, this paper introduces a use of confidence scores to weight words in the history, a weight of the prior topic distribution and a way of calculating perplexity that accounts for recognition errors in the model context.

1. INTRODUCTION

The task of a language model can be understood as calculating the probability $P(w_i|h_i)$, where w_i is the i-th word in the given text and h_i is a history of the word w_i, $h_i = w_1, \ldots, w_{i-1}$.

The most common type of language models used in speech recognition is n-gram language model. This model is simple and fast to use while giving good results. The basic assumption inherent to n-gram model is that a word's probability depends only on $n-1$ previous words instead of the whole word sequence starting at the beginning of the text. In other words $P(w_i|h_i) \approx P(w_i|w_{i-n+1}, \ldots, w_{i-1})$

While this assumption does not hold, this model has shown to be very powerful in practise and very hard to beat. Its main disadvantage is that realistically it cannot account for long-range dependencies of words. The range of dependencies is limited to n words where in practise n is small and rarely exceeds four. With small values of n, the model is less likely to hit the data sparsity problem.

Another feature inherent to n-gram model is that it bases its probability estimations on the frequencies of n-grams and ignores information beyond short n-gram i.e. syntax or semantics of the text. There has been a host of models introduced in the literature that aim to exploit this additional information; e.g. [5, 6].

The experience shows that it is important to train a language model on a text that matches the test set. For two texts to match, it is important that they come from the same or very similar sources and they cover related topics. Thus to improve a language model, it may help to incorporate topic information. This is the aim of so called topic-based language models.

There have been various language models exploiting topic information introduced in the literature. There are for example mixtures of topic specific n-grams [2], latent semantic analysis (LSA) based model [5] or a topic-based language model introduced by Gildea and Hofmann [1].

This paper introduces several extensions to the original model [1]; namely the weight b in (6) and the confidence weighting (7). The original model is summarised in sections 2.1 and 2.2. Section 2.3 introduces a way of integrating a long-span language model to a decoder.

Moreover, it has been reported [1] that a Probabilistic Latent Semantic Analysis (PLSA)-based model does not lead to WER reduction despite gains in perplexity. This paper proposes the use of recognised transcript as word history in perplexity calculation. Usually perplexity is calculated on correct transcripts but these do not contain errors made by a recogniser. We believe that calculating perplexity with recognised transcript as word history predicts WER better than the standard perplexity using word history from a reference transcript. Perplexities on two different training and test sets are presented in section 3.

2. PLSA-BASED LANGUAGE MODEL

PLSA is a general machine learning technique for modelling the co-occurrences of events. It uses a hidden variable that attains a finite number of values. In the PLSA framework, these values are referred to as aspects.

In the case of language modelling, the PLSA method models the co-occurrence of words and documents. At the intuitive level, each value from the hidden variable's dynamic range can be interpreted as a topic. The PLSA model in this paper is a mixture of unigram distributions. Each distribution corresponds to one aspect.

2.1. TRAINING A PLSA MODEL

The first step in building a PLSA-based language model [1] is making a co-occurrence matrix. One row of this matrix corresponds to one word from the language

model's vocabulary and one column corresponds to one document from the training corpus. One element of this matrix is the number of occurrences of the given word in the given document. This matrix is very sparse.

The model is estimated by the Expectation-Maximisation (EM) training. This uses the statistics from the co-occurrence matrix to find $P(z_k|d_i)$ and $P(w_j|z_k)$ that maximise the log-likelihood (1).

$$log\mathcal{L} = \sum_{i=1}^{N}\sum_{j=1}^{M} n(d_i, w_j) \log \sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k) \quad (1)$$

where $n(d_i, w_j)$ is the number of the occurrences of the word w_j in the document d_i, z_k is the k-th aspect, N is the number of documents in the training collection, M is the number of words in the vocabulary and K is the number of aspects or topics in the model.

The EM algorithm operates iteratively maximising the training data likelihood. The model parameters are initialised randomly. Each EM iteration comprises of two steps: the E-step and the M-step. For the given objective function (1), the E-step is

$$P(z_k|d_i, w_j) = \frac{P(z_k|d_i)P(w_j|z_k)}{\sum_{k=1}^{K} P(z_k|d_i)P(w_j|z_k)} \quad (2)$$

and the M-step is

$$P(w_j|z_k) = \frac{\sum_{i=1}^{N} n(d_i, w_j)P(z_k|d_i, w_j)}{\sum_{j=1}^{M}\sum_{i=1}^{N} n(d_i, w_j)P(z_k|d_i, w_j)} \quad (3)$$

$$P(z_k|d_i) = \frac{\sum_{j=1}^{M} n(d_i, w_j)P(z_k|d_i, w_j)}{n(d_i)} \quad (4)$$

2.2. CALCULATING WORD PROBABILITIES

A language model needs to calculate probabilities of words given their histories. Thus the question is how to use the probabilities from (3) and (4) in a language model.

The probabilities of the topics $P(z_k|d_i)$ are used as mixture weights when calculating the word probability. The history h_i of the current word is used instead of d_i to re-estimate these weights on the test set.

There are two cases to be distinguished. For the first word of a document, the topic distribution defaults to the distribution observed in the training data.

$$P(z_k|h_1) = P(z_k) = \frac{\sum_{w,d} n(w, d)P(z_k|d)}{\sum_{w,d} n(w, d)} \quad (5)$$

For all the other words in the document, the history is used to adapt the topic distribution to the current document. Formally, the topic distribution is defined as

$$P(z_k|h_i) = \frac{1}{i+1}\frac{P(w_i|z_k)P(z_k|h_{i-1})}{\sum_{q=1}^{K} P(w_i|z_q)P(z_q|h_{i-1})} + \frac{i}{i+1}P(z_k|h_{i-1}) \quad (6)$$

In the real world there are recognition errors in the history h_i therefore it is desirable to take into account the reliability of the tokens. Also at the beginning of a document, there is not enough information available to the model about the topic of the document. Thus we suggest using confidence scores (CSs) and emphasising the prior topic distribution in (6) to cope with these two issues.

$$P(z_k|h_i) = \frac{1}{i+b}\frac{(P(w_i|z_k)P(z_k|h_{i-1}))^{cs(i)}}{\sum_{q=1}^{K}(P(w_i|z_q)P(z_q|h_{i-1}))^{cs(i)}} + \frac{i-1+b}{i+b}P(z_k|h_{i-1}) \quad (7)$$

where $cs(i)$ is the confidence score of the i-th word. The parameter b is the weight of the prior topic distribution. For the first b words, the prior topic distribution has a higher weight than the estimate based on the current document. After seeing the first b words, the estimate based on the current document receives a higher weight.

Now the probabilities from (5) and (7) can be used to calculate the word probabilities $P(w_i|h_i)$ where

$$P(w_i|h_i) = \sum_{k=1}^{K} P(w_i|z_k)P(z_k|h_i) \quad (8)$$

A big advantage of this language model is that it can account for the whole document history of a word irrespective of the document length. On the other hand, it does not have means for representing the word order as this model is a mixture of unigram distributions. A remedy to this shortcoming is to combine the PLSA-based model with an n-gram . The model tested in this paper uses the way suggested by Gildea and Hofmann [1]:

$$P(w_i|h_i) \propto P_{n-gram}(w_i|h_i) * \frac{P_{PLSA}(w_i|h_i)}{P_{unigram}(w_i)} \quad (9)$$

2.3. USING PLSA IN DECODING

The PLSA-based language model's advantage - long-span history - constitutes a challenge when using it in decoding. A standard Viterbi algorithm-based decoder requires that the word history is short. This history is limited to $n-1$ words for an n-gram model. A language model that accounts for the whole history from the start of the document up to the current word cannot merge paths through the lattice. For each word with multiple histories, there must be a separate node that is assigned a language model probability according to its history. We decided to integrate the PLSA-based language model into a two-pass decoder to cope with this issue.

In the first pass, the decoder outputs a transcript with a confidence score [4] for every token and one lattice for every segment (utterance). At this stage, the decoder uses an n-gram model. In the second pass, the lattices are re-scored. During the re-scoring, the PLSA history comprises of all segments in a document but the current segment. This means that the PLSA history is fixed for all

words in a given segment. This "history" is not a language model history in the usual sense because it contains both past and future words. We refer to such a "history" as context (ctx). The definition of the n-gram history does not change.

3. EXPERIMENTAL EVALUATION

To evaluate the PLSA-based language model, perplexity was calculated on CTS reference transcripts. Two test sets were used: the test set used in NIST's Hub5 speech-to-text evaluation 2002 (eval02) and the test set from NIST's Rich Transcription Spring 2003 CTS speech-to-text evaluation (eval03). These test sets contain conversations from Switchboard and Fisher corpora released by LDC $www.ldc.upenn.edu$ [8].

3.1. RESULTS ON SWITCHBOARD CORPUS

In the first instance, the model was tested on the eval02 test set. This test set comprises of transcripts from Switchboard I and II databases. It has 62k words 19k of which come from Switchboard I.

The baseline model in the following perplexity tests was a 4-gram language model interpolated with a class-based trigram [3] model. This baseline was trained on CTS and broadcast news transcripts (500M words altogether). The CTS transcripts included 3M words of Switchboard I data, 200k words of Call Home English data, and 800k words of Switchboard II data. The class-based model had 350 classes trained on CTS transcripts. The broadcast news data included transcripts made by a company called PSM, transcripts released by LDC for the TDT task, transcripts from CNN's website and conversational-style data collected from the Internet [7].

The baseline model was part of Cambridge University's system produced for the NIST 2003 Rich Transcription evaluation. This model used Good-Turing discounting, modified Knesser-Ney smoothing and comprised of linearly interpolated component n-gram models.

The baseline's PP on eval02 was 61.7. When the baseline n-gram was combined with a PLSA model, the perplexity fell to 57.3 (7.2% relative reduction). The PLSA component had 750 aspects and was trained with 100 EM iterations. The PLSA model was trained on the same data set as the class-based trigram. Documents in the PLSA model were conversations.

Most of the data in the training set of the PLSA model was from Switchboard I database. Thus, it is interesting to see how the model performed on the Switchboard I subset of eval02. The baseline PP was 73.3 on this subset. Adding the PLSA model reduced PP by 9.9% to 66.1. These numbers show that PLSA reduces perplexity of a well trained n-gram . The reduction is greater if PLSA's training text relates to the test set.

In order to obtain a realistic assessment of the PLSA-based model, we calculated perplexity using the context definition from section 2.3. When using the reference transcript as the context, the test set perplexity fell to 55.9 (9.4% reduction). On the Switchboard I subset only, perplexity was 64.8 (11.7% reduction). However, during speech recognition only an erroneous recognised transcript is available for context. When using such a transcript as the context, the PP reduction was 3.7%. After employing (7), the reduction was 4.3%. Table 1 suggests that $b = 10$ is the best value out of the three included. $b = 1$ corresponds to the original PLSA model (6). $b = 100$ is too big especially for Switchboard I. Also the table shows that the use of confidence scores makes the PLSA model less sensitive to the value of b.

Model	swbI	swbIIp3	swbIIC	eval02
baseline	73.3	57.3	57.3	61.7
history,10	66.1/9.9	53.4/6.8	54.2/5.3	57.3/7.2
ref. ctx,10	64.8/11.7	51.9/9.5	53.0/7.5	55.9/9.4
rec. ctx,1	69.2/5.7	57.1/0.4	58.0/-1.2	60.9/1.4
rec. ctx,10	67.9/7.4	55.7/2.9	56.5/1.3	59.4/3.7
rec. ctx,100	68.5/6.6	55.2/3.7	56.0/2.2	59.3/4.0
+CSs,1	68.4/6.8	55.6/3.1	56.5/1.4	59.5/3.6
+CSs,10	68.0/7.3	55.1/4.0	56.0/2.2	59.1/4.3
+CSs,100	69.1/5.8	55.4/3.4	56.2/1.8	59.6/3.5

Table 1: Perplexity on eval02 and its subsets: Switchboard I, II phase 3, and II Cellular. The numbers after "/" are relative reductions in percent. The numbers after "," are values of b.

3.2. RESULTS ON FISHER CORPUS

Fisher is a new database of CTS data released by LDC in 2003/04. The training data of the n-gram from section 3.1 did not contain any Fisher data. Thus, a new n-gram component trained on 20M words of Fisher transcripts was added to make a fair baseline. These new 20M words were also added to the class-based model's training set and the number of classes increased to 500. The interpolation weights of the word and class-based n-grams were set manually to minimise perplexity of the PLSA-based model on one half of eval03 (eval03dev).

The test set perplexities were calculated on the remaining half of eval03 (eval03tst). The eval03 test set contains 36k words of Fisher data and 38k words of Switchboard II phase 5 data. Both eval03dev and eval03tst contained equal portions of Fisher and Switchboard II phase 5 subsets. The PLSA model had 750 aspects trained with 100 EM iterations starting from random values. It was observed that taking sides as documents lead to a slight improvement thus side-based documents were used.

PPs with various values of b were calculated to determine a suitable value of b. Tables 2 and 3 summarise PPs for $b = 100$ and $b = 10$. They show that $b = 100$ is better

in this case. It is interesting that the suitable values of b differ so much for eval03tst and eval02. This indicates that the PLSA model needs much more data to estimate the topic of Fisher conversations than it needs to find the topic on a Switchboard I conversation. This may relate to the differences in the methodology of collecting the data or to the different definitions of documents.

The baseline perplexity on eval03tst was 55.0. Adding the PLSA module pushed perplexity down to 51.2 which is 8.7% PP reduction. On the Fisher subset of eval03tst, this reduction was 8.9%.

To evaluate using different PLSA contexts on eval03tst, several test set perplexities were computed. The test showed that using the reference transcript as the context from section 2.3 yielded the PP of 48.1 on eval03tst which is a 12.6% improvement over the baseline. Thus, having a long context is very important and it makes up for the fact that words from the current segment are ignored in the context.

Model	swbIIp5	fsh04	eval03tst
baseline	56.9	52.9	55.0
history	52.1/8.5	48.2/8.9	50.2/8.7
ref. context	49.9/12.3	46.1/12.9	48.1/12.6
recognised ctx	55.9/1.8	50.3/4.9	53.2/3.3
rec. ctx.& CSs	54.8/3.8	49.6/6.2	52.3/4.9

Table 2: Perplexity on eval03tst and its two subsets: Switchboard II phase 5 and Fisher 2004. The numbers behind "/" are relative reductions in percents. b=100

Using (7) leads to the eval03tst perplexity of 52.3 and to a perplexity of 49.6 on the Fisher subset (6.2% and 4.9% reduction, respectively). Thus, the recognition errors can be compensated partly by using CSs.

Table 3 shows the importance of b. When b is set to 10, PLSA degrades the language model with recognised context if confidence scores are not used. However, when b is set to 100, using PLSA improves the language model. Moreover, comparing Tables 2 and 3 shows that using CSs makes the model less sensitive to the value of b.

The WERs of the recogniser that produced the recognised transcripts is lower on swbIIp5 than on fsh04. Tables 2 and 3 show that the relative gain with the recognised transcript is higher on fsh04. The gains with the reference transcript are about the same. This indicates that as the recogniser achieves lower WERs, PLSA will be more helpful. The same observation holds on eval02.

4. CONCLUSIONS AND FUTURE WORK

Perplexity tests have shown that employing PLSA with the suggested modifications in a language model for CTS reduces perplexity relative to state-of-the-art language models even when using recognition output as the context. This conclusion holds across two different test sets.

In the near future, we are going to re-score lattices to calculate WERs for our model. In a more distant future, it will be interesting to combine our primarily semantics-oriented model with a mainly syntax-based language model. In addition, PLSA for higher-order n-grams may bring further improvements.

Model	swbIIp5	fsh04	eval03tst
baseline	56.9	52.9	55.0
history	56.7/0.4	52.6/0.5	54.7/0.5
ref. context	53.0/7.0	49.1/7.2	51.1/7.1
recognised ctx	61.1/-7.3	54.8/-3.5	58.0/-5.5
rec. ctx.& CSs	56.9/0.0	51.8/2.0	54.5/1.0

Table 3: Perplexity on eval03tst and its two subsets: Switchboard II phase 5 and Fisher 2004. The numbers behind "/" are relative reductions in percents. b=10

5. ACKNOWLEDGEMENTS

This work was supported by DARPA grant MDA972-02-1-0013. The paper does not necessarily reflect the position or the policy of the US Government and no official endorsement should be inferred.

6. References

[1] Gildea, D. and Hofmann, T., "Topic-based language models using EM", Proc. of Eurospeech, 6, 1999.

[2] Moore, G.L. and Young, S.J., "Class-based Language Model Adaptation Using Mixtures of Word-class Weights", ICSLP, 2000.

[3] Young, S. Evermann, G. Kershaw, D. Moore, G. Odell, J. Ollason, D. Valtchev, V. Woodland, P., "The HTK Book" (for HTK Version 3.2), Cambridge University Engineering Department, 2002

[4] Evermann, G. and Woodland, P.C., "Large Vocabulary Decoding and Confidence Estimation Using Word Posterior Probabilities", Proc of ICASSP, 2000.

[5] Bellegarda, Jerome R., "Exploiting Latent Semantic Information in Statistical Language Modeling", Proc. of IEEE, Vol. 88, Num. 8, pp. 1279–1296, August 2000.

[6] Wang, W. and Harper, M.P., "Language Modeling Using a Statistical Dependency Grammar Parser", Proc. of ASRU, 2003.

[7] Bulyko, I. Ostendorf, M. and Stolcke, A., "Getting More Mileage from Web Text Sources for Conversational Speech Language Modeling using Class-Dependent Mixtures", To Appear in Proc. of HLT. 2003.

[8] Cieri, C. Miller, D. and Walker, K., "From Switchboard to Fisher: Telephone Collection Protocols, their Uses and Yields", Proc. of Eurospeech, 2003.