A COMPARISON OF DIALOGUE-STATE DEPENDENT LANGUAGE MODELS

Frank Wessel, Andrea Baader, and Hermann Ney

Lehrstuhl für Informatik VI, RWTH Aachen – University of Technology
Ahornstraße 55, 52056 Aachen, Germany
wessel@informatik.rwth-aachen.de

ABSTRACT

Dialogue-state dependent language models in automatic inquiry systems can be employed to improve speech recognition and understanding. In this paper, the dialogue state is defined by the set of parameters contained in the system prompt. Using this knowledge, a separate language model for each state can be constructed.

In order to obtain robust language models we study the linear interpolation of all dialogue-state dependent language models and an automatic text clustering algorithm. In particular, we extend the clustering algorithm so as to automatically determine the optimal number of clusters. These clusters are then be combined with linear interpolation.

We present experimental results on a Dutch corpus which has been recorded in the Netherlands with a train timetable information system in the framework of the ARISE project [1]. The perplexity, the word error rate, and the attribute error rate can be reduced significantly with all of these methods.

1. INTRODUCTION

If the choice of words a speaker is uttering correlates with the state a dialogue system is in, this knowledge can be used to improve the language model of the recognizer. In [2] and [6] the dialogue state is defined by the question the user is replying to. Using this definition, the language model training corpus is split according to the dialogue states and a separate language model for each dialogue state is then trained.

One of the main drawbacks of this approach is that the number of words in the language model training corpus for each dialogue state is rather small and that several dialogue states even remain unobserved in the training material. Possible ways to overcome this problem are to generalize dialogue states until a sufficient amount of training material for each state is obtained [2, 7] or to decide between the dialogue-state dependent and a global language model [6], if the first is not robust enough.

Both methods do not take into account that a single dialogue-state dependent language model can well contribute to the prediction of the user utterance in several other different dialogue state. It might thus be desirable to use a combination of all dialogue-state dependent language models for each dialogue state. In [9] we therefore proposed to train a language model for each dialogue state and use a linear interpolation of all dialogue-state dependent and a global language model for each dialogue state instead of deciding between the dialogue-state dependent and the global language model. In doing so, supplementary information contained in the different language models can be exploited.

In this paper we use the automatic text clustering algorithm presented in [3] to merge dialogue states until a sufficient amount of training material for each generalized dialogue state is obtained and compare this approach with our previous experiments, based on a linear interpolation of all dialogue-state dependent language models for each dialogue state. In addition, we extend the clustering algorithm so as to automatically find the optimal number of clusters using Leaving-One-Out.

Using a combination of the clustering algorithms and the linear interpolation, the number of generalized dialogue-state dependent language models can first be reduced and then be interpolated linearly as suggested in [9], thus speeding up recognition.

Parts of the corpus have been provided by KPN Royal Dutch Telecom. The responsibility for this study lies with the authors.

Table 1: Specification of the Dutch corpus

	training	testing
dialogues	7756	453
sentences	73402	4330
words	290745	18491

2. DESCRIPTION OF THE CORPUS

The corpus which we used for our experiments was recorded with the prototype of a Dutch train timetable information system in the framework of the ARISE project (see Table 1). The language model training material is identical to the transcriptions of the user utterances. The vocabulary used throughout all of the following experiments consists of 985 words, the phoneme inventory of 36 phonemes. Since we did not have access to the online version of the information system we ran all experiments off-line. For our experiments, we generated a word graph on the testing corpus with our own large vocabulary continuous speech recognition system [4].

3. DEFINITION OF THE DIALOGUE STATES

As in [2] and [6] we define the dialogue states in a natural way. In order to generate a database query, the system has to fill several slots and has to prompt questions to the user. Typically, the user will answer these questions in the desired way and provide the necessary information. In our case, the slots which have to be filled before a database query can be started are *station of departure*, *station of arrival*, *date* and *time*. With the four different slots defined above, $2^4 - 1 = 15$ potential dialogue states have to be considered. In addition, the system is capable of asking whether the user wants a repetition of the connection which has been retrieved from the database, whether he wants an earlier or later connection or whether he would like to obtain a completely different one. In combination, the system prompt can contain 19 different sets of parameters which can either be part of a question for this set or a verification of it. An additional garbage state is defined to enable a classification of dialogue states which obviously resulted from errors within the system.

We split the corpus according to the dialogue state of each utterance and thus obtain a separate training corpus for each dialogue state. We observed 32 of the 39 possible dialogue states in the language model training and 25 in the testing corpus. For the rest of this paper we will use the following notation: let S denote the number of different dialogue states, s the current dialogue state, $C_s = (w_{s,1} \cdots w_{s,N_s})$ the language model training corpus for dialogue state s and N_s the number of words in this corpus.

4. DEFINITION OF THE LANGUAGE MODELS

Let $N_s(h, w)$ denote the frequency of event (h, w) in training corpus C_s, $n_{0,s}(h)$ the number of different words which have not been observed after history h and W the size of the vocabulary. For each dialogue state s we constructed a trigram language model with the dialogue-state dependent training corpus C_s. The models for each dialogue state are based on absolute discounting. For smoothing, the relative frequencies are discounted with a discounting weight b_s and are interpolated with a generalized singleton backing-off probability distribution $\beta_s(w|\overline{h})$. Details are described in [5].

$$\begin{aligned} p_s(w|h) &= \max\left\{0; \frac{N_s(h,w) - b_s}{N_s(h)}\right\} \\ &+ b_s \cdot \frac{W - n_{0,s}(h)}{N_s(h)} \cdot \beta_s(w|\overline{h}) \quad . \end{aligned} \tag{1}$$

5. INTERPOLATION OF THE LANGUAGE MODELS

As described above, the motivation for this combined model is to investigate whether other dialogue states can contribute to the prediction of what the user is going to say. p_0 denotes the probability distribution provided by the global language model which has been trained on the whole training corpus and $\lambda_s(i)$ the interpolation weight for dialogue-state dependent language model i in dialogue state s:

$$\widetilde{p}_s(w|h) = \sum_{i=0}^{S} \lambda_s(i) \cdot p_i(w|h) \quad , \tag{2}$$

$$\text{where} \quad \sum_{i=0}^{S} \lambda_s(i) = 1 \quad \forall s \quad .$$

The main problem with this model is the rather large number of $(S+1)^2$ interpolation weights. In order avoid optimization on the testing data we would have had to split the training corpus into two parts using one of them for the training of the language models and the other as a cross-validation set for the estimation of the interpolation weights. This would have further deteriorated the language models. Instead, we decided to use the training corpus itself for the estimation of the $\lambda_s(i)$. Using the Expectation-Maximization-Algorithm for the estimation of the interpolation weights on these data would have lead to setting $\lambda_s(s) = 1$ and $\lambda_s(i) = 0 \quad \forall s \neq i$. Therefore we computed Leaving-One-Out probabilities on the training corpus and used these probabilities in the iteration formula:

$$\begin{aligned} p_s(w|h) &= \max\left\{0; \frac{N_s(h,w) - 1 - b_s}{N_s(h) - 1}\right\} \\ &+ b_s \cdot \frac{W - n_{0,s}(h)}{N_s(h) - 1} \cdot \beta_s(w|\overline{h}) \quad , \end{aligned} \tag{3}$$

where $\beta_s(w|\overline{h})$ and $n_{0,s}(h)$ are also modified accordingly. The modification of these quantities is very convenient in our language model software, since we store the counts of trigrams, bigrams and unigrams and compute the language model probabilities when needed. For details, the reader should refer to [8].

6. AUTOMATIC DIALOGUE-STATE CLUSTERING

In order to compare the interpolation of dialogue-state dependent language models with the generalization of dialogue states, we applied an automatic text clustering algorithm to merge the training corpora of several dialogue states until a sufficient amount of training material for each language model is obtained. We used the automatic text clustering algorithm presented in [3]. Let $\mathcal{K}$ denote a set of clusters, i.e. generalized dialogue-states. For simplification we define a mapping function $\mathcal{M}$ which maps a dialogue state s to one of the $|\mathcal{K}|$ generalized dialogue states:

$$\begin{aligned} \mathcal{M} : \mathcal{S} &\longrightarrow \mathcal{K} \\ \mathcal{M}(s) &\longmapsto k \quad . \end{aligned} \tag{4}$$

The criterion which has to be maximized is the log-likelihood F of the cluster dependent unigram language models $p_{\mathcal{M}(s)}(w)$ over all possible mapping functions which map a dialogue state s to a cluster $\mathcal{M}(s)$:

$$F = \sum_{s=1}^{S} \sum_{n=1}^{N_s} \log p_{\mathcal{M}(s)}(w_{s,n}) \tag{5}$$

The cluster-dependent unigram probability distribution for a specific cluster k in Equation (5) is defined as follows:

$$p_k(w) = \frac{N_k(w)}{N_k} \quad , \tag{6}$$

$$\text{where} \quad N_k(w) = \sum_{s:\mathcal{M}(s)=k} N_s(w)$$

$$\text{and} \quad N_k = \sum_{s:\mathcal{M}(s)=k} N_s \quad .$$

The clustering process is started by assigning one of the clusters to each dialogue state s at random. As in [3] we then perform an exchange algorithm, trying each cluster for each dialogue state. Finally the dialogue state is assigned to the best fitting cluster.

Although this algorithm works favourably well in terms of reducing the perplexity, the number of clusters cannot be determined automatically with the unigram perplexity criterion. Ideally, the clustering algorithm should choose that number of clusters which minimizes the perplexity on new, previously unseen data. For a number of clusters larger than or equal to the number of dialogue-states, the clustering algorithm presented in [3] will assign each dialogue-state to a distinct cluster, thus minimizing the perplexity on the training data. An automatic reduction of the number of clusters is not possible with this algorithm.

In order to simulate unseen data we therefore extend the clustering algorithm and use a different criterion. Instead of minimizing the log-likelihood F of the cluster dependent unigram models we now minimize the log-likelihood F of the cluster dependent Leaving-One-Out unigram models:

$$\begin{aligned} p_k(w) &= \max\left\{0; \frac{N_k(w) - 1 - b_k}{N_k - 1}\right\} \\ &+ b_k \cdot \frac{W - n_{0,k}}{(N_k - 1) \cdot W} \quad . \end{aligned} \tag{7}$$

Using this probability distribution in Equation (5) we perform the same exchange algorithm. Initially, the number of clusters is identical to the number of distinct dialogue-states, assuring that each cluster contains only one dialogue-state dependent language model training corpus. Using the conventional unigram models in Equation (5) the algorithm would stop at this point. With the Leaving-One-Out unigram models the algorithm now merges dialogue-states until no corpus is moved any more and the Leaving-One-Out perplexity on the training data is minimized.

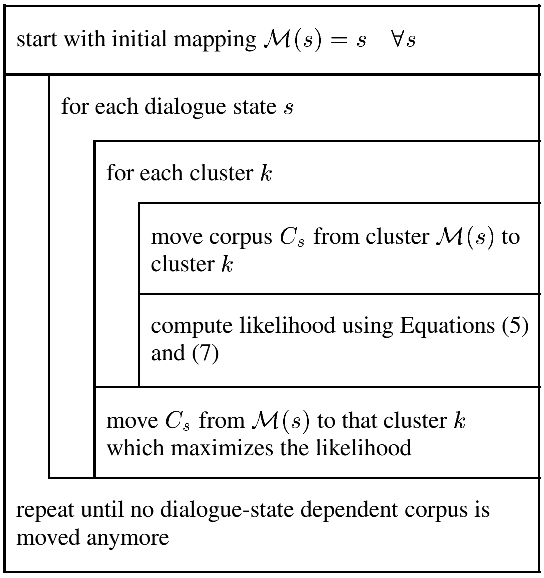

Figure 1: Text clustering algorithm used in combination with Leaving-One-Out unigram probabilities.

Table 2: Baseline word error rates for the global language model and the interpolation of the 32 dialogue-state dependent language models with the global language model.

trigram model	perplexity	errors [%] del / ins / WER
global	11.8	2.0 / 2.6 / 14.0
interpolated	9.3	1.8 / 2.5 / 13.2

7. EXPERIMENTAL RESULTS

In order to evaluate the performance of the different language models we measured the perplexities and the word error rates on the word graph. The graph error rate of the word graph we used is 7.2%. Table 2 comprises the baseline results achieved with the global language model and the interpolated model defined in Equation (2).

In a first experiment we used the automatic text clustering algorithm presented in [3]. We studied the effect of different numbers of clusters on the word error rate. For each cluster we trained a separate trigram language model. All dialogue states which were clustered together were then mapped to this generalized language model during the rescoring of the word graph. As Table 3 clearly indicates, the clustering algorithm can be used successfully to reduce the number of language models. The last line in Table 3 shows the performance of the dialogue-state dependent language models without any clustering. The impact on the word error rate is disappointing. The interpolated model in Table 2 performs significantly better than the generalized dialogue states in Table 3.

The main disadvantage of the automatic text clustering algorithm is that the optimal number of clusters (optimal in terms of reducing the perplexity on new data) cannot be determined automatically. As described before, we therefore changed the clustering criterion in order to incorporate Leaving-One-Out. Figure 2 summarizes the experiments carried out with this algorithm. Starting with a number of clusters equal to the number of distinct dialogue states, the algorithm reduces the number of clusters from 32 to 21 in a single iteration step and stops. The

Table 3: Perplexities and word error rates for different clusters computed with the unigram perplexity criterion. The language models are not interpolated.

number of clusters	perplexity	errors [%] del / ins / WER
5	9.8	1.9 / 2.8 / 13.6
10	9.8	2.0 / 2.6 / 13.6
15	9.9	2.0 / 2.6 / 13.7
20	10.0	2.0 / 2.6 / 13.7
25	10.0	2.0 / 2.6 / 13.7
30	10.0	2.1 / 2.6 / 13.7
32	10.0	2.1 / 2.6 / 13.7

upper of the two figures also shows the perplexities for a smaller number of clusters. We simply fixed this number and ran the algorithm in order to verify that the Leaving-One-Out unigram perplexity for 21 clusters is in fact minimal. We also computed the normal unigram perplexities for the the same clustering algorithm, shown in the lower of the two figures. These perplexities do of course increase with a decreasing number of clusters. Table 4 lists the distinct dialogue states which were clustered together with the Leaving-One-Out clustering criterion. As the table shows, the composition of the different clusters corresponds closely to what one might expect, e.g. question and verification turns are not clustered together.

In a final experiment we compared the global model, the interpolated model defined in Equation (2), and an interpolation of the 21 cluster language models with the global language model. The parameters of the last two models were estimated as described in [9]. As Table 5 shows, the last method does not improve the word error rate any further. The main advantage, though, is the smaller number of language models which have to be interpolated. Instead of 33 models, only 22 models are now interpolated and the computing time can thus be reduced.

For these three combined models we also measured the attribute error rate, defined as the number of incorrectly recognized attributes. An attribute in this context is determined by the database slot type, e.g. station of arrival, and the attributed value, e.g. station Amsterdam. The slot type and value were extracted from the best sentence of the recognizer output.

8. CONCLUSION

We presented experiments with dialogue-state dependent language models on a Dutch database which has been acquired with an automatic train timetable information system in the European ARISE project. We compared two methods for constructing robust dialogue-state dependent language models which are based on automatic text clustering on the one hand and the linear interpolation between several of the dialogue-state dependent and a global model on the other. Our experiments indicate that the parameters can be estimated reliably using Leaving-One-Out probabilities on the training corpus. A combination of both methods performs best in terms of attribute error rate and computing time.

In particular, we described a means of creating dialogue-state dependent language models automatically from a language model training corpus by using Leaving-One-Out for the clustering algorithm and for the estimation of the interpolation weights. No additional parameters have to be optimized manually.

With the combined model the attribute error rate has been reduced by 5% relative, from 14.7% with a dialogue-state independent language model to 14.0% with our best dialogue-state dependent model.

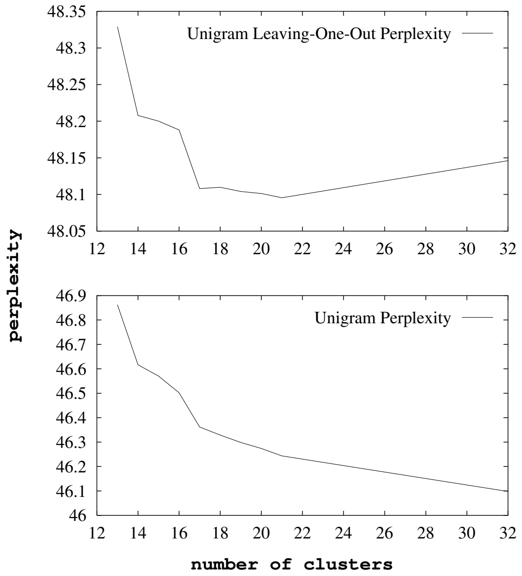

Figure 2: Leaving-One-Out unigram and normal unigram perplexities for different numbers of clusters computed with the Leaving-One-Out unigram perplexity criterion.

Table 4: Clusters obtained with the Leaving-One-Out clustering criterion.

Q: date
Q: station of arrival
GARBAGE, Q: date, time
Q: repeat connection
Q: station of arrival, date, Q: station of departure
Q: station of departure and arrival
Q: station of arrival, date, time, Q: station of departure and arrival, date, time
Q: previous train, Q: next train
Q: time
Q: new connection
V: date
V: date, time
V: repeat connection
V: station of departure, V: station of departure, date
V: station of arrival, V: station of arrival, date
V: station of departure and arrival
V: station of arrival, date, time, V: station of departure, date, time, V: station of departure and arrival, date, V: station of departure and arrival, date, time
V: station of arrival, time, V: station of departure and arrival, time
V: station of departure, time
V: previous train, V: next train
V: time

Table 5: Word and attribute error rates for selected language models

trigram model	perplexity	errors [%] del / ins / WER	errors [%] del / ins / AER
global	11.8	2.0 / 2.6 / 14.0	2.3 / 5.1 / 14.7
interpolated	9.3	1.8 / 2.5 / 13.2	2.1 / 5.6 / 14.0
clustered	9.2	1.9 / 2.5 / 13.2	2.0 / 5.6 / 14.0

9. ACKNOWLEDGEMENTS

The authors would like to thank Andreas Kellner from the Philips Research Laboratories in Aachen for his support in measuring the attribute error rates for the different language models.

10. REFERENCES

[1] P. Baggia, J.L. Gauvain, A. Kellner, G. Perennou, C. Popovici, J. Sturm, F. Wessel: ‘Language Modelling and Spoken Dialogue Systems - the ARISE experience’, to appear in Proc. Sixth European Conference on Speech Communication and Technology, Budapest, Hungary, September 1999.

[2] W. Eckert, F. Gallwitz, H. Niemann: ‘Combining Stochastic and Linguistic Language Models for Recognition of Spontaneous Speech’, in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1996, Atlanta, USA, pp. 423-426, May 1996.

[3] S. Martin, J. Liermann, H. Ney: ‘Adaptive Topic-Dependent Language Modeling Using Word-Based Varigrams’, in Proc. Fifth European Conference on Speech Communication and Technology, Rhodes, Greece, pp. 1447-1450, September 1997.

[4] H. Ney, L. Welling, S. Ortmanns, K. Beulen, F. Wessel: ‘The RWTH Large Vocabulary Continuous Speech Recognition System’, in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1998, Seattle, USA, pp. 853-856, May 1998.

[5] H. Ney, S. Martin, F. Wessel: ‘Statistical Language Modeling Using Leaving-One-Out’, in ‘Corpus Based Methods in Language and Speech Processing’, S. Young, G. Bloothoft (eds.), pp. 174-207, Kluwer Academic Publishers, The Netherlands, 1997.

[6] C. Popovici, P. Baggia: ‘Specialized Language Models Using Dialogue Predictions’, in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1997, Munich, Germany, pp. 815-818, April 1997.

[7] C. Popovici, P. Baggia, P. Laface, L. Moisa: ‘Automatic Classification of Dialogue Context for Dialogue Prediction’, in Proc. of the ICSLP 1998, Sydney, Australia, pp. 397-400, December 1998.

[8] F. Wessel, S. Ortmanns, H. Ney: ‘Implementation of Word Based Statistical Language Models’, in Proc. SQEL (*Spoken Queries in European Languages*) Workshop on Multi-Lingual Information Retrieval Dialogues, Pilsen, Czech Republic, pp. 55-59, April 1997.

[9] F. Wessel, A. Baader: ‘Robust Dialogue-State Dependent Language Modeling Using Leaving-One-Out’, in Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing 1999, Phoenix, USA, pp. 741-744, March 1999.