INITIAL LANGUAGE MODELS FOR SPOKEN DIALOGUE SYSTEMS

Andreas Kellner

Philips GmbH Forschungslaboratorien Aachen
Weisshausstrasse 2; 52066 Aachen, Germany
kellner@pfa.research.philips.com

ABSTRACT

The estimation of initial language models for new applications of spoken dialogue systems without large task-specific training corpora is becoming an increasingly important issue.

This paper investigates two different approaches in which the task-specific knowledge contained in the language understanding grammar is exploited in order to generate n-gram language models for the speech recognizer: The first uses class-based language models for which the word-classes are automatically derived from the grammar. In the second approach, language models are estimated on artificial corpora which have been created from the understanding grammar.

The application of *fill-up* techniques allows the combination of the strengths of both approaches and leads to a language model which shows optimal performance regardless of the amount of training data available.

Perplexities and word error rates are reported for two different domains.

1. INTRODUCTION

Now that spoken dialogue systems have become mature enough to leave the research labs and to hit the market, it becomes increasingly important to quickly develop new applications. However, the stochastic models used in a speech recognizer usually require the collection and transcription of large amounts of training data. Our experience shows that the acoustic features for a new application may in general be trained on data from different applications in the same language. Unfortunately, this is not the case for the stochastic language model which still needs a lot of task-specific data.

In many spoken dialogue systems like the one described in Section 2, a manually constructed grammar is used to parse the output of the speech recognizer for application-specific information. This paper investigates different approaches to exploit the task-specific knowledge contained in this grammar in order to generate an n-gram language model for the recognizer. Special emphasis is put on the start-up situation when no or only very little training material is available.

Section 2 describes our architecture for spoken dialogue systems and the underlying stochastic model. In Section 3, the grammar is used to automatically create classes for class-based language models. Section 4 examines n-gram language models which are directly created from the stochastic context-free grammar (SCFG). Section 5 proposes fill-up techniques which manage to combine the benefits of both approaches. Experimental results in two different domains are reported in Section 6.

2. SYSTEM ARCHITECTURE

Figure 1 shows the Philips Inquiry Systems Architecture which provides a generic framework for spoken dialogue systems [1, 6].

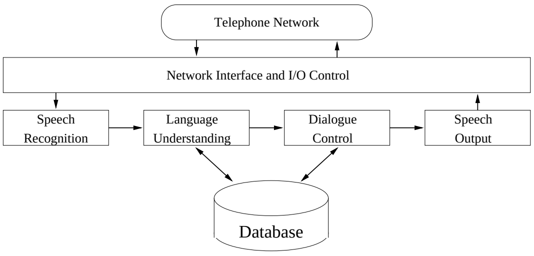

Figure 1: System architecture.

The speech recognizer creates a *word graph* which contains multiple sentence hypotheses and their acoustic likelihood $P(O|W)$. The subsequent language understanding module computes the best path through this word graph and derives its meaning. It uses a stochastic context-free grammar (SCFG) which covers the meaningful parts of the input, the so-called *concepts* [1]. The remainder of the input is modeled by a separate *filler model*, e.g. a word n-gram. Figure 2 shows an excerpt from such a SCFG for our automated exchange-board task PADIS [6]. The numbers in

This work was partially funded by the German Federal Ministry for Education, Science, Research and Technology (BMBF) in the framework of the Verbmobil Project under Grant 01 IV 701 C9. The responsibility for the contents of this study lies with the authors.

parentheses are the rule probabilities which have to be estimated on a task specific corpus. The understanding module's language model $P_{\text{und}}(W)$ used to determine the best path through the word graph consists of two parts:

```
<request_type>  ::= (0.39)  the <info_type> of
<request_type>  ::= (0.61)  a connection to

   <info_type>  ::= (0.40)  phone_number
   <info_type>  ::= (0.12)  address
   <info_type>  ::= (0.27)  email
   <info_type>  ::= (0.21)  fax_number
```

Figure 2: Example of SCFG rules

1) The stochastic grammar, which provides the probability $P(\tilde{w}_i|k_x)$ that the word sequence $\tilde{w}_i$ is used to realize a given concept k_x.
2) A concept bigram language model which models the order of the concepts in the user's utterance.

Even though the word graph, which is the output of the speech recognizer does not contain any language model scores, an internal language model $P_{\text{rec}}(W)$ (e.g. a bigram) has to be used to focus the search and to allow for real-time operation. The remainder of this paper describes the problem of estimating an initial $P_{\text{rec}}(W)$ for a new domain.

3. CLASS-BASED LANGUAGE MODELS

When an initial language model has to be estimated for a new application of a spoken dialogue system, there is usually not enough data for the robust training of a bigram or trigram model. It is well known that in such situations, the robustness of an n-gram can be increased by using class models [3]. Under the assumption that each word w belongs to exactly one class $c(w)$, this leads to

$$P(w_2|w_1) = P(w_2|c(w_2)) \cdot P(c(w_2)|c(w_1))$$

One problem in many language modeling tasks is to define an appropriate classification $w \rightarrow c(w)$ [7]. In a spoken dialogue system, however, the user's input is often well structured and some classes based on the semantics of a word (e.g. numbers or names) can be derived manually (cf. [4, 8]).

Furthermore, it is possible to derive the mapping $w \rightarrow c(w)$ automatically from the SCFG:
A context-free grammar usually contains some non-terminal symbols which may be expanded to many different terminal symbols (single words). All these words may then be grouped into one class. From the example in Figure 2, a new class which contains the words '*phone_number*', '*address*', '*email*', and '*fax_number*' would be created by this approach. All words which are not assigned to a class in this way form a class on their own. This approach automatically generates classes for words which are expected to appear in the same word context (e.g. different station names, or numbers).

The results in Section 6 show that the class-based bigram language model performs significantly better than a word-based bigram, when a small amount of training data is available.

4. LANGUAGE MODELS DERIVED FROM THE GRAMMAR

In many cases, there might not even be enough training data for the robust training of class-based language models. In the start-up situation, the only task-specific information we have is the one coded in the understanding grammar.

In principle, a stochastic grammar can directly be used as language model in the recognizer. However, when the language model should be integrated with other stochastic language models, it is desirable to derive a regular n-gram from the SCFG. The precise computation of n-grams from a SCFG is described in [10]. A much simpler approach is to apply Monte-Carlo methods in order to create an artificial corpus from the stochastic grammar. This corpus can then be used for the estimation of n-gram language models [5].

Our results show that even an artificial corpus created from an untrained grammar can be used to generate a bigram language model which performs one order of magnitude better in perplexity than the competing zerogram and more than 20% better in terms of word error rate.

When the rule probabilities of the SCFG and the concept bigram can be estimated on a small amount of real training data, even more realistic pseudo-corpora can be generated. Table 1 shows some statistics of a real corpus collected in a field test of the PADIS system [6] in comparison to artificial corpora whose generating grammars were trained on different amounts of data.

Size of training set	SCFG generated corpora				Real
for SCFG (#sent.)	0	100	1000	10,000	corpus
#concepts/sent.	13	3.0	2.1	1.8	1.8
#words/concept	2.8	1.2	1.5	1.5	1.6

Table 1: Statistics for artificial and real corpora.

The perplexity as well as the word error rate decrease quickly when the underlying grammar and concept bigram are trained on a small amount of application data (cf. Section 6). The main benefit comes from the concept language model. Its vocabulary contains only about 20 different concepts. It can therefore be estimated robustly on a very small database.

Unfortunately, the performance of this type of language model can not be improved further when a large amount of training data is available.

This in part is due to the fact that our SCFG covers only the meaningful parts of the input but not the *fillers* which make up about 10% of the words in the real corpus. 645 of the 1996 words in the recognizer's lexicon are not used as terminal symbols of the grammar and therefore do not appear in the artificial corpora at all.

5. COMBINATION OF DIFFERENT APPROACHES

While language models trained on artificial corpora perform very well in the start-up situation with only a few sentences of training data, class-based language models seem to be the best choice as soon as a certain amount of realistic data is available. It is therefore desirable to combine the strengths of both approaches into one stochastic language model which performs well regardless of the size of the training corpus.

A standard approach for combining different stochastic language models is to use linear interpolation of the models. Here a different technique was used:

Two or more language models can be combined in a hierarchical way using *fill-up* models. In [2], this was used to combine a speaker-dependent language model with a general fall-back model. The fill-up technique proved to be better suited for the task than regular linear interpolation methods.

In the *fill-up* approach, one language model $P_1(W)$ is used at the top-level, a second model $P_2(W)$ acts as *fall-back* model. If an n-gram was not seen in the training of $P_1(W)$, its likelihood is derived from the fall-back model $P_2(W)$.[1] The theory is discussed in [2].

For the creation of word language models for a new spoken dialogue system, we have three different stochastic models which can be combined:

1) The word language model **W** which has been trained directly on the corpus.
2) The class-based language model **C** (Section 3).
3) The language model **G** which has been derived from the grammar (Section 4).

Different combinations of these models have been investigated. A two-level model consisting of the model trained on an artificial corpus **G** and the class bigram **C** showed the best result for most applications.

In the final model, **G**→**C**, the grammar bigram **G** is used as top-level model. When a bigram history was not found in this model (e.g. for *filler phrases*), the class language model **C** is used as fall-back model.

6. EXPERIMENTAL RESULTS

To evaluate the performance of the different language models, experiments in various domains were carried out. This section shows the results for two different tasks: An automatic train-timetable information (TABA) [1] and an automatic telephone switchboard (PADIS) [6].

For both applications, some partial training-sets were created by using only the first n utterances of the whole training corpus. The performance of the language models estimated on the basis of these training sets was evaluated on a separate test set.

Table 2 shows the perplexity (PP) and word error rate (WER) of the different models in the TABA domain.

As discussed before, the bigram trained on the artificial corpus shows the best *initial* performance but is not as good as the other models for *large training corpora*. The performance of the fill-up model **G**→**C** (cf. Section 5) is also listed in Table 2.

A direct comparison of the combined model and the simple models can be found in Figure 3.

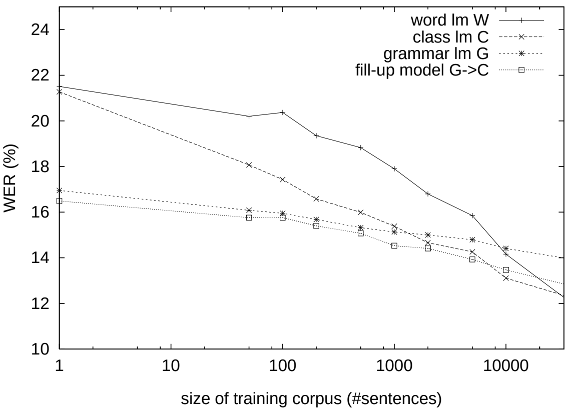

Figure 3: Word error rates for the TABA system

The same experiments were carried out in the PADIS domain (see Figure 4). Here, one can see an additional effect:
The standard word bigram shows the best performance for large training sets. This is due to a strong correlation between adjacent words (e.g. first-name and last-name) which is not modeled by the class language models or the models created from the artificial corpus. In some cases it may not be desirable to model such relations which occur in the training data (e.g. when there is a known mismatch between the inquiries in the data collection and the real application of a a system). When such a relation has to be included in the n-gram model, or when any relation between different information items in a sentence is explicitly known

[1] This approach may be used recursively leading to multi-level models.

training set		word lm		class lm		grammar lm		$G \rightarrow C$	
#sent	#words	PP	WER	PP	WER	PP	WER	PP	WER
0	0	2974	21.51%	2486	21.27%	364	16.95%	251	16.49%
50	226	148	20.20%	82	18.07%	71	16.08%	60	15.76%
100	470	121	20.37%	75	17.43%	61	15.95%	60	15.76%
200	855	93	19.35%	53	16.58%	52	15.68%	51	15.40%
500	2067	78	18.83%	45	15.99%	47	15.32%	47	15.07%
1000	4024	65	17.90%	38	15.38%	44	15.13%	43	14.53%
2000	8033	51	16.80%	33	14.66%	41	15.00%	40	14.41%
5000	20416	38	15.85%	28	14.26%	36	14.79%	35	13.93%
10000	41199	27	14.16%	21	13.11%	32	14.41%	27	13.47%
33081	110216	18	12.26%	17	12.35%	27	13.99%	22	12.85%

Table 2: Comparison of models for TABA

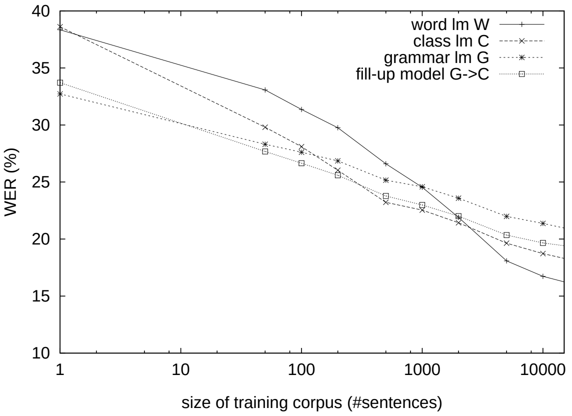

Figure 4: Word error rates for the PADIS system

(e.g. from a telephone database), this information can easily be integrated in the generation of pseudo corpora using the stochastic framework proposed in [9]. This was, however, not yet tested.

7. CONCLUSION

The results reported in this paper showed that good language models can be estimated from very little training data, if the task-specific information contained in the stochastic language understanding grammar is exploited. Language models trained on artificial corpora perform reasonably well without any task-specific training data. When a small amount of training data is available, class-based language models (for which the classes can automatically be derived from the grammar) show the best performance.

To combine the benefits of both approaches, a two-level fill-up language model was introduced. This approach led to the best performance for training corpora of realistic size.

8. ACKNOWLEDGEMENT

The author would like to thank Jochen Peters, Reinhard Kneser, Dietrich Klakow and Stefan Besling for many interesting discussions on the subject.

9. REFERENCES

[1] H. Aust, M. Oerder, F. Seide, and V Steinbiss. The Philips automatic train timetable information system. *Speech Communication*, 17(3–4):249–262, Nov. 1995.

[2] S. Besling and H.-G. Meier. Language model speaker adaptation. In *Proc. EUROSPEECH*, pages 1755–1758, Madrid, Spain, Sep. 1995.

[3] A. M. Derouault and B. Merialdo. Natural language modeling for phoneme-to-text transcription. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8:742–749, November 1986.

[4] Sunil Issar. Estimation of language models for new spoken language applications. In *Proc. ICSLP*, volume 2, pages 869–872, Philadelphia, PA, October 1996.

[5] D. Jurafsky, et. al. Using a stochastic context-free grammar as a language model for speech recognition. In *ICASSP*, volume 1, pages 189–192, Detroit, MI, May 1995.

[6] A. Kellner, B. Rueber, and F. Seide. A voice-controlled automatic telephone switchboard and directory information system. In *Proc. IVTTA*, pages 117–120, Basking Ridge, NJ, Sep. 1996.

[7] H. Ney, U. Essen, and R. Kneser. On structuring probabilistic dependencies in stochastic language modelling. *Computer Speech and Language*, 8:1–38, 1994.

[8] C. Popovici and P Baggia. Language modelling for task oriented domains. In *Proc. EUROSPEECH*, volume 3, pages 1459–1462, Rhodes, Greece, September 1997.

[9] F. Seide, B. Rueber, and A. Kellner. Improving speech understanding by incorporating database constraints and dialogue history. In *Proc. ICSLP*, volume 2, pages 1017–1020, Philadelphia, PA, Oct. 1996.

[10] A. Stolcke and J. Segal. Precise n-gram probabilities from stochastic context-free grammars. In *Proceedings of ACL*, pages 74–79, 1994.