TOWARDS BETTER LANGUAGE MODELS FOR SPONTANEOUS SPEECH

B. Suhm†, *A. Waibel* †,‡

† Carnegie Mellon University (USA)
5000 Forbes Avenue
Pittsburgh PA 15213, USA

‡ Karlsruhe University (Germany)
Am Fasanengarten 5
76131 Karlsruhe, Germany

ABSTRACT

In our effort to build a speech–to–speech translation system for spontaneous spoken dialogs we have developed several methods to improve the language models of the speech decoder of the system. We attempt to take advantage of natural equivalence word classes, frequently occuring word phrases, and discourse structure. Each of these methods was tested on spontaneous English, German and Spanish human–human dialogs.

1. INTRODUCTION

The goal of the JANUS project is multi-lingual machine translation of spontaneously spoken dialogs in a limited domain: two people scheduling a meeting with each other. We are currently working with English, German, and Spanish as source languages and German, English, and Japanese as target languages. Table 1 shows the size of training and test set for the English, German and Spanish Spontaneous Scheduling Task databases (ESST, GSST, SSST) used for all experiments reported in this paper, and the coverage of the dictionary over the test set. [1]

	ESST	GSST	SSST
# Training Dialogs	136	192	96
# Training Tokens	43 K	44 K	36 K
# Test Dialogs	14	10	10
Coverage	98%	95%	94%

Table 1: Training and Test Set for ESST, GSST and SSST

Spontaneous speech is challenging to recognize, because of acoustic and grammatical disfluencies, which are frequent especially in spoken human–human dialogs. In addition, the collection and transcription of spontaneous dialogs is a costly and time consuming process. Therefore, the development of language models for tasks like Scheduling has to take into account the highly ungrammatical and disfluent nature of word sequences and the very limited amount of training data in form of transcripts.

First, we report on how we applied and extended an automatic word clustering algorithm which naturally finds classes of words optimized for perplexity reduction. Then, we propose a word phrase bigram language model to reduce perplexity as well as automatically define common word phrases or idioms in a given task. Finally, we describe an attempt to take advantage of semantic and discourse knowledge by dynamically interpolating a set of sublanguage models trained on specific concepts or speech–acts.

2. CLUSTER BASED LANGUAGE MODELS

Word class based language models have been suggested in the literature [1, 2, 3, 4]. Instead of estimating the probabilities of word N–grams [2] $P(w_i \mid w_{i-1})$ a word class based model assigns each word w_i to a class $C(w_i)$ and estimates probabilities of class N-grams $P(C(w_i) \mid C(w_{i-1}))$ and class membership distributions $P(w_i \mid C(w_i))$. The number of classes is typically one to two orders of magnitude smaller than the number of unique words in a task, thus reducing the number of free parameters of the language model. Therefore, word class based models are particularly useful for training on smaller amounts of data.

The main issue for a word class language model is how the assignment of words to classes is found. Parts–of–Speech have been used and lead to perplexity reductions. In a different approach, Jelinek [1]

[1] A training token can either be a regular word or a noise word. Dialogs typically consist of 8–10 utterances, and an utterance of 20–50 tokens. The coverage is the percentage of the words in the test set that also occur in the training set.

[2] In this paper we consider only bigram probabilities for simplicity and due to the fact the higher order models typically need more training data than is available for spontaneous databases.

proposed a method to find nuclear parts–of–speech based on a mutual information criterion.

We applied another approach proposed by Kneser [4] which learns word equivalence classes automatically to minimize the perplexity of the resulting class bigram model. In a clustering algorithm, the (arbitrary) initial assignment of words to classes is changed to minimize the perplexity on independent data. A step in the clustering involves moving one word to a new cluster and checking the leaving–one–out perplexity [5] as optimization criterion. This criterion allows training data to be used as independent data by simulating unseen events.

We extended the method by applying a simulated annealing technique to prevent the clustering from relaxing into a local optimum. Instead of moving a word to a new cluster only if resulting in a decrease of the leaving–one-out perplexity, we allow an increase which is determined by the annealing function and converges to zero with increasing number of iterations.

We generated cluster bigram models on ESST, GSST, SSST and the ATIS data of November '92. For all databases, the number of clusters proved to be a quite robust parameter of the model, with a (very flat) optimum around 150 clusters. Table 2 compares the perplexities for the word bigram, word trigram and cluster bigram model.

	ESST	GSST	SSST	ATIS
Word Bigrams	38	82	74	20
Word Trigram	36	75	67	15
Cluster Bigrams	39	84	73	20
Word & Cluster Bigrams	35	73	66	19

Table 2: Perplexities of Word–N–Gram and Cluster Bigram Model

On our Scheduling databases, the linear interpolation of the cluster with the word bigram model reduces perplexity significantly compared to the word bigram model, and only very slightly when compared to the word trigram model. For the ATIS databases with an order of magnitude more language model training data, the cluster model doesn't yield improvements.

Across languages, the clusters represent in part semantic or syntactic categories such as weekdays, months, numbers and conjunctions, or expressions with similar meaning such as to be busy/available and to express consent/disagreement. This observation inspired applications of the clustering beyond language modeling to corpus–based learning of translation of spontaneous speech.

3. WORD PHRASE BASED LANGUAGE MODEL

3.1. MOTIVATION AND APPROACH

Looking at the transcripts we noticed that certain word phrases are very frequent. For instance in our English database, the most frequent word sequences include "I'll see you then", "I'll meet you from", "I'm out of town", "in the afternoon", "in the morning".

To take advantage of this observation we propose to build a language model which bundles sequences of words into frequently occuring phrases. Our approach is to build a bigram model where the tokens can be both single words and sequences of words. For instance for the sentence "I'll be out of town", instead of looking at the word bigrams "I'll be", "be out", "out of" and "of town" one may consider the bigrams "I'll be" "be out_of_town".

However, introducing word sequences as additional dictionary entries will in general increase the sparseness of the training data and thus deteriorate the quality of the probability estimates for the bigrams. Therefore, word phrases may not be arbitrarily included in the model. Instead, we suggest to determine the word sequences automatically in an optimization algorithm towards reducing test set perplexity.

3.2. THE WORD PHRASE FINDING ALGORITHM

The structure of the optimization algorithm is very similar to the algorithm Kneser [4] suggested to automatically cluster words. The basic idea is to improve the optimization criterion by making local optimizations. After choosing a sequence of words as a word phrase candidate, we create a language model including the new word phrase and determine its leaving–one–out perplexity. Word phrase candidates which don't decrease perplexity are removed from the dictionary of the language model, before the next iteration begins with choosing the next candidate.

The selection of word phrase candidates is crucial to make the algorithm computationally efficient: with dictionaries in the order of thousands of words there are already millions of possible word pairs. We suggest to choose the candidates according to their mutual information, i.e. we consider the pair of tokens t_i, t_j as next candidate to form a word phrase $t_i\_t_j$ which has maximal mutual information

$$log \frac{P(t_i, t_j)}{P(t_i)P(t_j)}$$

among all possible pair of tokens, according to the current dictionary.

A closed formula for the leaving–one–out logprob

$$LP_{LO} = \sum_{i=1}^{N} log P(t_i \mid t_{i-1})$$

where N is the number of tokens t_j in the training set, can be derived similarly to the cluster bigram model, which is described in detail in [4]. However, to have a meaningful comparison between the perplexity of a regular word bigram and a word phrase bigram model, words within a phrase have to be counted as separate words when computing perplexity. Therefore, we need to normalize LP_{LO} by the number of *words* W occuring in the training set and use

$$PP_{LO} = e^{-\frac{1}{W} LP_{LO}}$$

instead of LP_{LO} as optimization criterion.

3.3. WORD PHRASE LANGUAGE MODELS

Table 3 shows perplexities for the word phrase models, their dictionary sizes and how many of the dictionary entries are actually word phrases. Compared to the word bigram model, the word phrase model yields modest perplexity reductions.

	ESST	GSST	SSST	ATIS Nov92
Perplexity	35	78	70	18
Dictionary Size	1246	2052	1416	1863
# Word Phrases	121	115	51	170

Table 3: Word–Phrase Bigram Model

Frequency of occurrence, mutual information and perplexity are closely related concepts in the context of word phrase finding. However, looking at the list of word sequences ordered by frequency of occurrence in the corpus, one notices that the optimization towards minimal leaving–one–out perplexity doesn't find just the most frequent sequences of words, since there is a complex interaction between lowering perplexity in the immediate context when forming a word phrase and deteriorating the probability estimates for bigrams which contain substrings of the word phrase.

4. STATISTICAL DIALOG MODELING

As recent studies [6] show, negotiations between human partners, as in our Scheduling domain, tend to be highly disfluent, both with respect to non–speech events and the structure of the utterances. However, significant constraints are embedded in semantics and discourse. We attempted to take advantage of these higher–level constraints in a language model which dynamically adapts according to current predictions about the dialog state, extending work by Pieraccini [7] and Ward [8].

We used output of the robust semantic parser PHOENIX [9] to automatically label different parts of a sentence with frames and slot fillers, and to extract the sequence of top–level slots and the corresponding sequences of words. A junk slot absorbs all words which are not covered by the semantic grammar.

Slot	Training	Dictionary	Perplexity
give-info	15 K	493	34
temporal	8 K	393	38
suggest-time	6 K	307	33
interject	6 K	327	17
junk	20 K	1059	50

Table 4: Slot Sublanguages for ESST

Then, we trained word bigram language models $P(w_{i+1} \mid w_i, S_i)$ for each slot S_i (which represents a semantic concept or speech–act), and a slot bigram model $P(S_{j+1}|S_j)$. Table 4 shows the size of the training text for each slot, the dictionary size and the test set perplexity for the sublanguage models of the most frequent slots. As can be seen, there is a significant variation in dictionary size and perplexity.

In the speech decoding process, we adapt the language model dynamically by interpolating the slot–dependent bigram models according to the current prediction of the next slot $P(S_i \mid S_{i-1})$. One can imagine the search for word sequences as a hidden Markov process with its states being the top–level slots, representing the current dialog state.

	ESST	GSST
Word Bigram	38	82
Static Interpol.	80	201
Dynamic Interpol.	37	80

Table 5: Perplexities for Slot–Dependent Models

Table 5 compares the perplexities of the dynamic slot–dependent model with the regular word bigram model and a model which interpolates the sublanguage models for each slot statically (i.e. each sublanguage model is assigned equal weight). As can be seen, the slot–dependent model doesn't yield significant improvements over the regular word bigram model. The high perplexity for the static interpolation shows that a good prediction of the next slot is crucial. However, the slot-dependent model reached a slot prediction accuracy of only about 55%. Improving the slot prediction by using a slot–trigram model was not possible due to sparsity of training data on our still fairly small spontaneous databases.

5. RECOGNITION RESULTS

We tested the described language models with the JANUS speech recognizer which is described in more detail in [10]. The acoustic modeling is based on a hybrid HMM and LVQ algorithm with a set of context–dependent phonemes and separate noise models. On the well known Resource Management task, this recognizer achieves a word accuracy of above 94%. Both the cluster and word phrase bigram model can be easily incorporated in a forward decoding pass of a time synchronous viterbi beam search.

	ESST	GSST
Word Bigrams	61%	59%
Cluster Bigrams	59%	59%
Word Phrases	66%	61%
Slot Dependent	60%	58%

Table 6: Preliminary Word Recognition Accuracies

The results are shown in Table 6. The cluster model did not improve recognition performance, but slightly degraded it, despite having reduced the task perplexity. The word phrase model improved recognition performance for all languages.

In addition, we tested the long range slot dependent model by rescoring lattice output of the recognizer in a A* search decoding pass. However, no improvements in word accuracy could be obtained.

As recent studies show [11, 12], human–human dialogs are significantly more disfluent than human–machine dialogs. Therefore, our human–human scheduling dialogs are very challenging to recognize, leading to significantly lower word accuracies than on well–known tasks like Wall Street Journal or ATIS.

6. CONCLUSION

We reported on three approaches to improve language models in spontaneous spoken dialog tasks, across several languages. A clustering algorithm which naturally finds classes of words reduced the perplexity of the resulting word class bigram model, but did not reduce the word error rate when incorporated in the speech decoder. However, the word classifications obtained can be used for other problems, e.g. automatically inferring grammars. The proposed word phrase finding algorithm can define common word phrases or idioms in a given task. The word phrase bigram language model obtained could both reduce perplexity and improve recognition performance. Extreme sparsity of training data for most of the sublanguages may be a reason that the approach to statistical dialog modeling didn't yield major improvements yet.

7. ACKNOWLEDGEMENTS

The authors gratefully acknowledge the help of the members of the JANUS project at CMU and University of Karlsruhe. The work was supported in part by grants from ARPA, NSF and ATR Interpreting Telecommunication Laboratories. We gratefully acknowledge their support.

8. REFERENCES

[1] F. Jelinek: *Self–Organized Language Modeling for Speech Recognition,* in A. Waibel, K.F. Lee: *Readings in Speech Recognition,* Morgan Kaufmann 1990, pp. 450–506

[2] P.F. Brown, P.V. deSouza, R.L. Mercer, V.J. Della Pietra, J.C. Lai: *Class–Based n-gram Models of Natural Language,* Computational Linguistics, Vol. 18, No. 4, pp. 467–479

[3] F. Pereira, N. Tishby, L. Lee: *Distributional Clustering of English Words,* Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics, 1993

[4] R. Kneser, H. Ney: *Improved Clustering Techniques for Class-Based Statistical Language Models,* EUROSPEECH 93, Berlin, Vol. 2, pp. 973–976

[5] H. Ney, U. Essen: *Estimating Small Probabilities by Leaving–One–Out,* EUROSPEECH 93, Berlin, Vol. 3, pp. 2239–2242

[6] B. Suhm, L. Levin, N. Coccaro, J. Carbonell, K. Horiguchi, R. Isotani, A. Lavie, L. Mayfield, C. P. Rosé, C. Van Ess-Dykema, A. Waibel: *Speech–Language Integration in a Multi–Lingual Speech Translation System,* AAAI Workshop on Integration of Speech and Natural Language Processing, Seattle, 1994

[7] R. Pieraccini, E. Levin, C.-H. Lee: *Stochastic Representation of Conceptual Structure in the ATIS Task,* Proceedings of the DARPA Speech and Natural Language Workshop, 1992

[8] W. Ward, S. Young: *Flexible Use of Semantic Constraints in Speech Recognition,* IEEE International Conference on Acoustics, Speech and Signal Processing, Minneapolis, 1993, Vol. 2, pp. 49–50

[9] W. Ward: *Understanding Spontaneous Speech: The Phoenix System,* IEEE International Conference on Acoustics, Speech and Signal Processing, 1991, Vol. 1, pp. 365–367

[10] M. Woszczyna, N. Aoki-Waibel, F.D. Buo, N. Coccaro, K. Horiguchi, T. Kemp, A. Lavie, A. McNair, T. Polzin, I. Rogina, C.P. Rose, T. Schultz, B. Suhm, M. Tomita, A. Waibel: *JANUS 93: Towards Spontaneous Speech Translation,* IEEE International Conference on Acoustics, Speech and Signal Processing, 1994, Vol. 1 , pp. 345–348

[11] S. Oviatt: *Predicting and Managing Spoken Disfluencies during Human–Computer Interaction,* to appear in Proceedings of the ARPA Human Language Technology workshop, Plainsboro, 1994

[12] B.Suhm: *Disfluencies in Spontaneous Human–Human and Human–Machine Dialogs,* CMU Technical Report (in preparation)