BMBF

Verbmobil

Verbundvorhaben

A CATEGORY BASED APPROACH FOR RECOGNITION OF OUT-OF-VOCABULARY WORDS

F. Gallwitz, E. Nöth, H. Niemann

F.–A.–Universität Erlangen–Nürnberg

Vm

Report 132
Mai 1996

Mai 1996

F. Gallwitz, E. Nöth, H. Niemann

Lehrstuhl für Mustererkennung (Inf. 5)
Friedrich–Alexander–Universität Erlangen–Nürnberg
Martensstr. 3
D–91058 Erlangen

Tel.: (09131) 85 - 7890
e-mail: gallwitz@informatik.uni-erlangen.de

Gehört zum Antragsabschnitt: 3.11, 3.12, 10.1

Das diesem Bericht zugrundeliegende Forschungsvorhaben wurde mit Mitteln des Bundesministers für Bildung, Wissenschaft, Forschung und Technologie unter dem Förderkennzeichen 01 IV 102 H/0 gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei den Autoren.

A Category Based Approach for Recognition of Out-of-Vocabulary Words

F. Gallwitz, E Nöth, H. Niemann

Lehrstuhl für Mustererkennung (Informatik 5), Universität Erlangen–Nürnberg,
Martensstraße 3, D-91058 Erlangen, Germany
email: `gallwitz@informatik.uni-erlangen.de`

ABSTRACT

In almost all applications of automatic speech recognition, especially in spontaneous speech tasks, the recognizer vocabulary cannot cover all occurring words. There is always a significant amount of out-of-vocabulary words even when the vocabulary size is very large. In this paper we present a new approach for the integration of out-of-vocabulary words into statistical language models. We use category information for all words in the training corpus to define a function that gives an approximation of the out-of-vocabulary word emission probability for each word category. This information is integrated into the language models. Although we use a simple acoustic model for out-of-vocabulary words, we achieve a 6% reduction of word error rate on spontaneous speech data with about 5% out-of-vocabulary rate.

1. INTRODUCTION

In almost all speech recognition applications out-of-vocabulary (OOV) words pose an important problem. In real time dictation applications, the user can control via the screen if a word was misrecognized. S/he can replace it with the correct word or extend the lexicon with the unknown word. In other applications like information retrieval over the telephone the user might not even know that the system misrecognized because of an OOV word. So if a user asks our train timetable inquiry system [3] "*I want to go from Sussex*" and *Sussex* is not in the lexicon, the system might recognize *I want to go at six* and it might respond with "*You want to leave at six o'clock. Where do you want to go?*". The user does not know, what the system understood and might react unpredictably. In addition to leading to a recognition error for itself, the OOV word often causes additional errors for the words that follow directly [11].

Thus it seems desirable to have a filler model that covers OOV words during the recognition process. Obviously, this is not an easy task: The filler model should in the ideal case cover *all* possible OOV words, which, by their nature, cannot be predicted in advance. Besides, it should *not* cover the words in the lexicon to avoid false alarms. The acoustic modeling of such filler models has been subject of several recent publications, mostly in the context of word-spotting applications [9, 8]. But obviously, language model information is similarly important to recognition of OOV words. For example, the probability of an OOV word following is comparatively high after the word sequence "*Hello, my name is*". The straightforward approach of substituting all OOV words in the language model training corpus by the label `OOV` before training an n-gram model has two basic drawbacks:

- The training corpus is usually taken into account when the vocabulary of a word recognizer in a specific application is determined. Conventionally, almost all words (except for word fragments and mispronounced words) in the training corpus are added to the vocabulary for optimal recognition performance. This leads to a drastic mismatch in the frequency of OOV words in the training corpus and in independent test sets. A solution to this problem called *Iterative Substitution* has been proposed in [5].
- A single `OOV` label for all OOV words cannot incorporate much language model information, because it has to cover fundamentally different classes of OOV words such as word fragments and proper names.

In this paper, we propose a solution to the problem of building language models for recognition of OOV words that is based on a system of word categories, which may be either constructed manually or automatically. We estimate emission probabilities of OOV words for each word category, which even allows us to provide category information on the OOV word that may be used by a parser. The resulting language model can easily be transformed into a word based model if necessary. In section 2 and 3 we describe the corpus based vocabulary design which is used in many applications, and the basic idea of category based n-gram language models, which build the framework for our approach. In section 4 we explain our method of integrating OOV words into a category based language model. In section 5 we show how we model OOV words on the acoustic level and present first results with the approach. In the last section we will conclude the paper with some remarks on future work.

2. CORPUS-BASED VOCABULARY DESIGN

A fundamental problem of designing word recognizers for all practical applications is the definition of an appropriate vocabulary. It should cover as much of future user utterances as possible. At the same time, it should not contain unnecessary words because they may lead to recognition errors, and computation time increases with growing vocabulary. For a detailed discussion of optimizing recognizer vocabulary see [11].

What system designers have to do before defining a vocabulary is to predict future user utterances as good as possible. The best way of predicting future user behavior is the observation of real users in the desired application. This is why multi-user speech recognition systems are often enhanced in a bootstrap procedure: The first version of the system contains a rather small vocabulary that may be based on wizard-of-oz experiments (e.g. [7]). The vocabulary may have been enhanced by an expert through adding a certain amount of words that seem useful for the application, e.g. completion of word categories or through adding inflections of observed words. Recognizer performance will certainly not be optimal in this state, because only little domain specific training data could be used, and because the out-of-vocabulary rate is rather high. Thus, it will be useful to record all user utterances to increase the amount of training data. After running the system for a certain time, the user utterances collected by the system can be transcribed. The vocabulary is now increased by those words in the corpus that seem useful for the application (again possibly modified by an expert). After retraining the system, the recognition performance should now be better than that of the previous version.

This gives us the following situation: We have a vocabulary V_{NEW} that was defined after the training corpus was observed and that will be used for our recognizer, and a basic vocabulary $V_{\text{BASIC}} \subseteq V_{\text{NEW}}$ that was defined without taking the training corpus of our language model into account. Any words that may have been in previous recognizer vocabularies that are not in V_{NEW} may be ignored. The vocabulary V_{BASIC} may also be empty, e.g. if the training corpus is sufficiently large before the first version of the recognizer is trained. This partition of the vocabulary will be essential for estimating OOV word probabilities in section 4.

Before we show how we use this information for calculating OOV word probabilities of word categories, we summarize some of the basic ideas of category based language models in the next section.

3. CATEGORY-BASED LANGUAGE MODELS

Category based n-gram models are a type of stochastic language model where words are pooled in categories or word classes. This can be done manually under linguistic aspects [2], or automatically (e.g. [10, 1, 4]). For the approach proposed in this paper we assume a disjoint category system. Each word w_i of a word sequence $\underline{w} = w_1 w_2 \ldots w_m$, then belongs to a unique category $c_i = c(w_i)$, and the conditional probabilities of a category-based n-gram are written as a product of the word membership score and a category n-gram probability.

$$\begin{aligned} P(\underline{w}) &= P(c_1) P(w_1 \mid c_1) \cdot \\ & \cdot \prod_{i=2}^{m} P(c_i \mid \underbrace{c_{i-n+1} \ldots c_{i-1}}_{n-1}) \cdot P(w_i \mid c_i) \end{aligned}$$

This type of language model can easily be transformed into a word based n-gram language model, e.g. for a trigram model:

$$P(w \mid uv) = P(c(w) \mid c(u)c(v)) \cdot P(w \mid c(w))$$

Transition probabilities $P(c_n \mid c_1 \ldots c_{n-1})$ between word categories can be estimated in the same way they are estimated for word based n-gram models: A maximum likelihood (ML) estimator

$$\widehat{P}(c_n \mid c_1 \ldots c_{n-1}) = \frac{\#(c_1 \ldots c_n)}{\#(c_1 \ldots c_{n-1})}$$

(where the function # counts the frequency of a given category sequence in the training corpus) is smoothed using suitable discounting, backing-off or interpolation strategies [12]. An estimation $\tilde{P}(w \mid c)$ of the word emission probabilities is obtained by smoothing the ML estimator $\widehat{P}(w \mid c)$ that is calculated by dividing the frequency $\#(w)$ of a word w in the training corpus by the frequency of all words v that are in the same category as w:

$$\widehat{P}(w \mid c) = \frac{\#(w)}{\sum_{\{v \mid c(v) = c(w)\}} \#(v)}$$

Our approach of estimating OOV probabilities does not assume any specific smoothing techniques. We do not reestimate the category transition probabilities, and for simplicity we reduce the smoothed word emission probabilities $\tilde{P}(w \mid c)$ linearly by a factor $(1 - \widehat{P}(\text{OOV} \mid c))$ where $\widehat{P}(\text{OOV} \mid c)$ is an estimation of the OOV probability in the word category c[1]. In the following section we will explain how we estimate $\widehat{P}(\text{OOV} \mid c)$.

4. ESTIMATION OF OOV PROBABILITIES

We will motivate our approach with the following example: Assume a manually constructed word category CITY, and a basic vocabulary V_{BASIC} (see section 2) that contains 500 city names. In the training corpus we have a total of 1000

[1] More sophisticated methods of reestimating emission probabilities seem possible. Especially for large values of $\widehat{P}(\text{OOV} \mid c)$ the OOV probability mass should be taken mainly from those words with few observations in the training corpus.

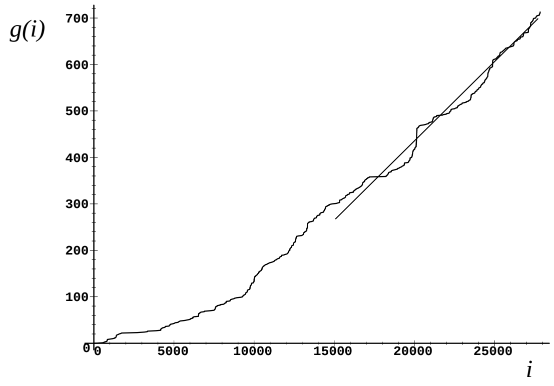

Figure 1: Estimation of the current OOV rate for the EVAR-system.

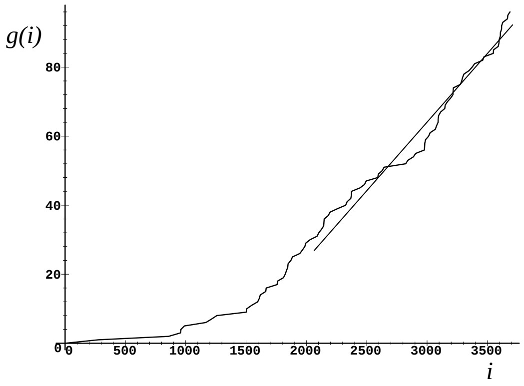

Figure 2: Estimation of the current OOV word probability for word category CITY.

occurrences of a word in category CITY. Fifty of these observations are city names that are not in V_{BASIC}, each of which appears once. We add the fifty new city names to the vocabulary V_{NEW} and calculate estimates $\tilde{P}(w \mid CITY)$ for every word w in category CITY. Obviously, it would be useful to take into account that actually 5% of our CITY-observations were OOV-words, and we have no reason to believe that this probability will be much lower for future test samples. But traditional methods of estimating language model probabilities ignore this, and the resulting emission probabilities for the 550 city names in V_{NEW} will sum up to 1.

We will now define a function that enables us to estimate the total OOV probability, or the OOV probability for arbitrary word categories. We will first define the framework for estimating the total OOV probability: Let w_i be the i-th word in the training corpus (corpus size t) and

$$L_i := \bigcup_{j=0}^{i} \{w_j\}$$

the set of all words that have been observed up to the i-th word of the training corpus. We now define the vocabulary V_i that we would have chosen if we had redefined our vocabulary after only observing the first i words of the training corpus. It would have contained all words of the basic vocabulary V_{BASIC}, and, additionally, all words that have been observed up to the i-th word of the training corpus that we integrated into V_{NEW}:

$$V_i := V_{\text{BASIC}} \cup (V_{\text{NEW}} \cap L_i)$$

This enables us to count observations of words that would have been out of vocabulary if we had redefined our vocabulary after each observed word:

$$f(i) := \begin{cases} 1, & \text{if } w_i \notin V_{i-1} \\ 0, & \text{else} \end{cases}$$

$$g(i) := \sum_{j=0}^{i} f(j)$$

If we now construct a linear approximation f in a local neighborhood of i, its slope gives us a good approximation of the OOV rate that corresponds to the vocabulary V_i. For $i = t$, we have an approximation of the expected OOV rate for V_{NEW}. The neighborhood should be large enough to be robust to local fluctuations of the OOV rate. On the other hand, it should not be too large to capture long term changes of the OOV rate. These can be due to the increasing vocabulary size, but also to changes in user behavior. Figure 1 shows the function g for the EVAR train timetable inquiry corpus (see section 5 for details). Although the vocabulary size increases from 1110 (V_{BASIC}) to 1558 (V_{NEW}), the OOV rate gets even higher. The reason for this is that in the beginning the users were typically friends of the system designers and were aware of the restricted capabilities of our system. When the telephone number of our system circulated via newspapers, the amount of users with very little knowledge on automatic speech recognition increased. The figure also shows a linear approximation of the part of g that corresponds to the more recent user utterances. Its slope 0.035 gives an estimation of the OOV rate we should expect when using V_{NEW} in the current version of our system (3.5%).

We can now use the same methods for estimating analogous functions that correspond to each category c of our language model. The same measures as described above are defined for each category, where w_i then gives us the i-th occurrence of a word from category c in the training corpus. Figure 2 shows the function g on the same corpus for word category CITY, which indicates a current OOV probability of about 2.4%.

For most of our manually constructed word categories the OOV probability is 0, because they describe a finite set of words. We have 5 word categories that are practically unlimited for our domain (e.g. REGION, SURNAME). Additionally we define categories for rare words that are not in other categories (OOV probability 73%) and for garbage (e.g. word fragments, OOV probability 100%).

5. EXPERIMENTS AND RESULTS

We used a SCHMM two-pass recognizer (Mel-cepstrum+Δ-features, 250 codebook classes, bigram+polygram language model) for the experiments described in this paper [6]. The acoustic model is rather simple: It is a 'flat' model that consists of a fixed number of HMM states with equal probability density functions. The number of HMM states was determined empirically on an independent validation sample. The probability density function is obtained by averaging over all frames in our training data that do not belong to silence- or noise-periods. The number of recognized OOV-words was tuned on the validation sample with a parameter that increases the acoustic score of the OOV-model HMM-states in each time frame.

Experiments were performed on the EVAR corpus of spontaneous speech data collected by our spoken dialog system [3, 13] which is able to answer inquiries about German Intercity train connections. The data were divided into a training sample, a validation sample and a test sample (Table 1). We used our initial V_{BASIC} vocabulary (1110 words) for the experiments to have a higher number of OOV words in the test sample. Using this vocabulary, the OOV rate in the test sample is 5.3%. The OOV probabilities were estimated as described in the previous section. For calculation of word error rates we substituted all occurrences of OOV words in the reference by `OOV`, which is the symbol also produced by the word recognizer for OOV words.

Without OOV models, the word error rate is 22.9%; with OOV models, it is 21.49%. This is a 6% word error rate reduction, although the OOV detection rate of about 15% is rather poor, and the false alarm rate of about 75% is very high. This has no negative effect on the word error rate because most of the false alarms are due to words that would have been misrecognized in any case.

6. CONCLUSION & FUTURE WORK

The method presented in this paper allows the estimation of language models for OOV words based on an arbitrary system of word categories. Although we used a rather primitive acoustic model for OOV words, we achieved a 6% reduction of word error rate on spontaneous speech data. Further improvements will be possible when enhanced acoustic models are used, e.g. phone- or syllable-grammars. We will also investigate more sophisticated techniques of taking into account OOV probabilities when word emission probabilities are estimated.

sample	phone calls	utter.	words	diff. words
training	804	7732	27852	1112
validation	54	441	1577	273
test	234	2383	8346	580

Table 1: Overview of training-, validation-, and test sample

ACKNOWLEDGEMENTS

The work presented in this paper was partly supported by the DFG (German Research Foundation) under contract number 810 830-0.

7. REFERENCES

1. J. R. Bellegarda, J. W. Butzberger, Y.-L. Chow, N. B. Coccaro, and F. Naik. A Novel Word Clustering Algorithm Based on Latent Semantic Analysis. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 172–175, Atlanta, GA, 1996.
2. H. Cerf-Danon and M. El-Beze. Three Different Probabilistic Language Models: Comparison and Combination. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 297–300, Toronto, 1991.
3. W. Eckert, T. Kuhn, H. Niemann, S. Rieck, A. Scheuer, and E.G. Schukat-Talamazzini. A spoken dialogue system for german intercity train timetable inquiries. In *Proc. European Conf. on Speech Technology*, pages 1871–1874, Berlin, 1993.
4. A. Farhat, J.-F. Isabelle, and D. O'Shaughnessy. Clustering Words for Statistical Language Models Based on Contextual Word Similarity. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 180–183, Atlanta, GA, 1996.
5. P. Fetter, A Kaltenmeier, T. Kuhn, and P. Regel-Brietzmann. Improved Modeling of OOV Words In Spontaneous Speech. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 534–537, Atlanta, GA, 1996.
6. F. Gallwitz, E.G. Schukat-Talamazzini, and H. Niemann. Integrating large context language models into a real time word recognizer. In *Proc. 3rd Slovenian-German and 2nd SPRS Workshop on Speech and Image Understanding*, Ljubljana, Slovenia, appears 1996.
7. L. Hitzenberger and H. Kritzenberger. Simulation Experiments and Prototyping of User Interfaces in a Multimedial Environment of an Information System. In *Proc. European Conf. on Speech Technology*, volume 2, pages 597–600, Paris, September 1989.
8. T. Kemp and A. Jusek. Modeling Unknown Words in Spontaneous Speech. In *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, volume 1, pages 530–533, Atlanta, GA, 1996.
9. H. Klemm, F. Class, and U. Kilian. Word- and Phrase Spotting with Syllable-Based Garbage Modelling. In *Proc. European Conf. on Speech Technology*, volume 3, pages 2157–2160, 1995.
10. R. Kneser and H. Ney. Improved Clustering Techniques for Class-Based Statistical Language Modelling. In *Proc. European Conf. on Speech Technology*, pages 973–976, 1993.
11. R. Rosenfeld. Optimizing Lexical and N-gram Coverage Via Judicious Use of Linguistic Data. In *Proc. European Conf. on Speech Technology*, volume 3, pages 1763–1766, 1995.
12. E.G. Schukat-Talamazzini. Stochastic language models. In *Electrotechnical and Computer Science Conference*, Portoroz, Slovenia, 1995.
13. E.G. Schukat-Talamazzini, T. Kuhn, and H. Niemann. Speech Recognition for Spoken Dialog Systems. In H. Niemann, R. De Mori, and G. Hahnrieder, editors, *Progress and Prospects of Speech Research and Technology*, number 1 in Proceedings in Artificial Intelligence, pages 110–120. Infix, 1994.