A Bootstrap Training Approach for Language Model Classifiers

V. Warnke, E. Nöth, J. Buckow, S. Harbeck, H. Niemann

University of Erlangen-Nuremberg
Chair for Pattern Recognition
Martensstr. 3, 91058 Erlangen, Germany
(warnke,noeth,buckow,snharbec,niemann)@informatik.uni-erlangen.de

ABSTRACT

In this paper, we present a bootstrap training approach for language model (LM) classifiers. Training class dependent LM and running them in parallel, LM can serve as classifiers with any kind of symbol sequence, e.g., word or phoneme sequences for tasks like topic spotting or language identification (LID). Irrespective of the special symbol sequence used for a LM classifier, the training of a LM is done with a manually labeled training set for each class obtained from not necessarily cooperative speakers. Therefore, we have to face some erroneous labels and deviations from the originally intended class specification. Both facts can worsen classification. It might therefore be better not to use all utterances for training but to automatically select those utterances that improve recognition accuracy; this can be done by a bootstrap procedure. We present the results achieved with our best approach on the VERBMOBIL corpus for the tasks dialog act classification and LID.

1. INTRODUCTION

Language models (LM) are very important for automatic speech recognition systems; they are widely used in word recognizers to estimate the probability of a word chain in order to reduce the number of possible paths in forward decoding or to find the best word chain in a word hypotheses graph or lattice. If LM are trained class dependent and run in parallel, they can serve as classifiers for tasks like topic spotting, dialog act classification, and language identification [9]. LM work on every kind of symbol sequence with a finite vocabulary, e.g. word sequences, phoneme sequences, or codebook class sequences; they are thus applicable to many domains, even if there is no word information available.

We use LM classifiers within the VERBMOBIL-project to classify and segment dialog acts. VERBMOBIL is a speech-to-speech translation project [1] in the domain of appointment scheduling and travel planning. Covered languages are German, English and Japanese. If, for instance, two persons try to fix a meeting date, time, and place, the system detects automatically the two languages of the partners and translates between them. To keep track of the dialog it is necessary for the system to know the state of the dialog at all times. This is done in terms of dialog acts (DA) as one of the tasks of the *dialog module* within VERBMOBIL. DA are, e.g., "greeting", "confirmation of a date", "suggestion of a place" [3]. Furthermore DA are used for template based translation. The VERBMOBIL-system works with a fall–back mechanism to improve the robustness of the system. If the deep linguistic analysis is not able to translate a turn in a predefined time interval, the recognized DA are used for template based translation. Thus it is at least possible to roughly translate the meaning of a turn if a detailed analysis failed. For our experiments we use the 18 DA from the first phase of VERBMOBIL which were defined based on their illocutionary force [3].

Because the training of LM is done with a manually labeled set of training data for each class obtained from not necessarily cooperative speakers, we have to face some erroneous labels and deviations from the topic specific to the given scenario. Both facts can worsen classification. It might therefore be a better way not to use all utterances for training of a LM but to automatically select only those utterances that improve recognition accuracy; this can be done by a bootstrap training.

In the following sections we describe our currently best approach for the bootstrap training of LM classifier.

2. MOTIVATION

In [6] the selection of words for a limited size dictionary for a large vocabulary speech recognition task is not done on the whole training corpus but rather on certain sub corpora. It is assumed that in the North American Business (NAB) application of read newspaper texts certain factors like time of the year in which the test corpus is collected has an influence on the words that should be in the dictionary. This should be represented in the training corpus which is used for designing the recognition dictionary. It is shown, that the best lexical coverage is achieved when only 19% of the training corpus were used.

In analogy to this approach we want to look at the question whether we can train better LM if we only use sub corpora of our training corpora. For the selection of the sub corpora we do not want to depend on heuristics but rather use cost functions and global optimization methods. Figure 1 shows the recognition results on a independent test set when we use randomly chosen subsets of our DA training corpus. As can be seen we sometimes have a slight decrease of the recognition rate, because the addition of "rarebirds" or erroneously labeled training tokens leads to worse probability estimates. Even with random selection the best results are not achieved with the complete training corpus.

[1]This work was funded by the German Federal Ministry of Education, Science, Research and Technology (BMBF) in the framework of the VERBMOBIL Project under Grant 01 IV 102 H/0. The responsibility for the contents lies with the authors.

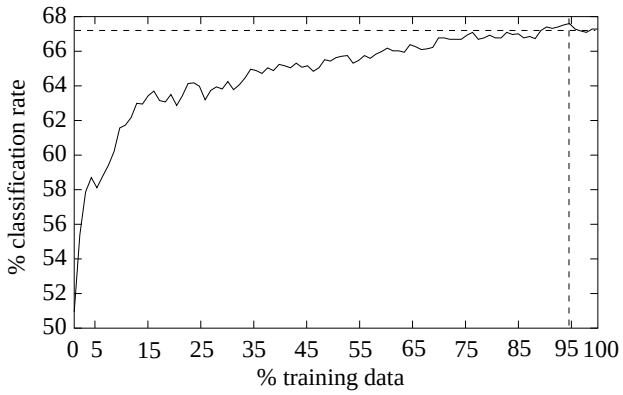

Figure 1: Recognition results for different number of training tokens with a LM classifier for 18 DA.

3. POLYGRAM LANGUAGE MODELS

In most cases language models are used to calculate the probability of a word sequence $\mathbf{w} = \omega_1 \dots \omega_T$ in a given language or context. We use *polygram language models* (PLM) [7] which are a special kind of *stochastic* n-gram model to estimate the probability of every kind *symbol sequence* where a symbol could be a word, phoneme or a codebook class.

3.1. Maximum Likelihood Estimation

Using polygrams the probability of the symbol sequence $\omega_1 \dots \omega_T$ is calculated with the help of

$$P(\mathbf{w}) = \prod_{i=1}^{T} P(\omega_i \mid \underbrace{\omega_1 \omega_2 \dots \omega_{i-1}}_{\text{history}}). \quad (1)$$

Because the younger history $\omega_{i-N+1} \dots \omega_{i-1}$ of the symbol sequence $\omega_1 \dots \omega_T$ is more important for modeling, and to restrict the number of free parameters inside the LM, we only use the last $N-1$ symbols to approximate the probability of $P(\mathbf{w})$:

$$\hat{P}(\mathbf{w}) = \prod_{i=1}^{T} P(\omega_i \mid \omega_{i-N+1} \dots \omega_{i-1}) \quad (2)$$

With this shorter history we can estimate the conditional probabilities $P(\omega_i \mid \omega_{i-N+1} \dots \omega_{i-1})$ from a given training corpus simply by using the maximum likelihood (ML) estimation:

$$P(\omega_i \mid \omega_{i-N+1} \dots \omega_{i-1}) = \frac{\#(\omega_{i-N+1} \dots \omega_i)}{\#(\omega_{i-N+1} \dots \omega_{i-1})}, \quad (3)$$

where $\#(\cdot)$ denotes the frequency of its argument in the training data. Of course, one would like to choose a large number of N for the history length – the approximation made by a LM of higher order gets closer to the real probability. Unfortunately, the number of parameters which have to be estimated increases exponentially with the size of N, and thus the ML estimates become far from being reliable because of the limited training data.

A compromise with respect to this trade-off between the model context N and the training data size can be made by introducing a weighted interpolation scheme.

3.2. Interpolation

The basic idea of applying interpolation methods is to fall back on the probability estimation of subsequences shorter than N. An example is the *linear interpolation* which uses all subsequences up to the length N [5] :

$$\begin{aligned} \tilde{P}(\omega_i \mid \omega_{i-N+1} \dots \omega_{i-1}) = \lambda_0 \cdot \frac{1}{L} \\ + \lambda_1 \cdot P(\omega_i) \\ + \lambda_2 \cdot P(\omega_i \mid \omega_{i-1}) \\ + \sum_{n=3}^{N} \lambda_n \cdot P(\omega_i \mid \omega_{i-n+1} \dots \omega_{i-1}). \end{aligned} \quad (4)$$

The fraction $1/L$ accounts for unseen sequences, where L is the number of words known to the LM, and ensures that no probability is set to zero. The interpolation coefficients λ_n can be estimated using the *Expectation Maximization (EM)* algorithm [7] on a given validation set.

Another interpolation method, which we do not describe in this paper, is the *rational interpolation*; it gives a higher weight to those n-grams seen very frequently in the training set. This method is described in detail in [8].

3.3. Language Models as Classifiers

After a LM has been trained for each of the considered classes, the models can be used to classify a symbol sequence. If we have the K classes $\Omega_1 \dots \Omega_K$, and P_k denotes the LM for class k, $k \in \{1, \dots, K\}$, the likelihood

$$P(\mathbf{w} \mid \Omega_k) = P_k = \prod_{i=1}^{T} \tilde{P}(\omega_i \mid \omega_{i-N+1} \dots \omega_{i-1}) \quad (5)$$

has to be computed for each class and the symbol sequence $\mathbf{w}$ is classified by computing the *a posteriori* probability using the Bayes' rule, where p_k is the *a priori* probability of class Ω_k and decide for k^* which has the maximum a posteriori probability:

$$k^* = \underset{k}{\operatorname{argmax}}(p_k \cdot P(\mathbf{w} \mid \Omega_k)). \quad (6)$$

Now we are able to classify each test utterances $\mathbf{w_j}$ of our test set $S = \{S_1, \dots, S_K\}$ into one of the K classes. Thus we can give the mean value of the classification result for each class ($\overline{RR}$) and the overall classification rate (RR):

$$\overline{RR} = \frac{1}{K} \sum_{k}^{K} \frac{\#(correct\ classified\ \mathbf{w}\ of\ \Omega_k)}{\mid S_k \mid} \quad (7)$$

$$RR = \frac{\sum_k^K \#(correct\ classified\ \mathbf{w}\ of\ \Omega_k)}{\sum_k^K \mid S_k \mid}, \quad (8)$$

where $\mid \cdot \mid$ returns the number of tokens in the set given as argument.

4. THE BOOTSTRAP TRAINING

The bootstrap training proceeds in two steps: first we have to split the training data into two disjunctive sets, the *training* set and the *validation* set and select randomly the first training token

split training data into training set TS and validation set VS				
select randomly the first training token for each class Ω_k from TS and train class dependent LM				
FOR all K classes				
	run all LM in parallel ; compute QF_k on VS and store it to LQF_k			
WHILE not all tokens of TS have been tested				
	FOR all K classes			
		FOR all **not** selected tokens of class Ω_k from TS		
			add current token to training data of class Ω_k and train new class dependent LM	
			run all K LM in parallel and compute QF_k for class k on VS	
			IF	$LQF_k \leq QF_k$
			THEN	mark current token as selected; continue with next class
			ELSE	reset LM to training state before last token; continue with next token
	FOR all K classes			
		run all LM in parallel; compute QF_k on VS and store it to LQF_k		
	IF	no improving token is selected for any class		
	THEN	stop		

Figure 2: Iterative bootstrap training procedure

for each class. Second we have to select iteratively the next best token from the training set that improves a *quality factor* QF (defined below) on the validation set. Figure 2 gives an informal account of the bootstrap procedure.

After the training data have been split into training TS and validation VS set, the first training token for each class dependent LM is selected randomly from the TS. Now we have to initialize the K accumulators LQF_k in which we store the computed QF from the *last* iteration. This is computed on the VS using the LM, trained on the first token as shown in Figure 2 (Block 3). For our experiments we used two different QF. The first is the class dependent recognition rate RR_k, which is computed by

$$RR_k = \frac{\#(correct\ classified\ \mathbf{w}\ of\ \Omega_k)}{\mid S_k \mid}. \qquad (9)$$

The second is the *perplexity* PX_k which is computed class dependent on the VS and could by retrieved using the formula

$$PX_k = exp(\frac{\sum_j^{|VS_k|} log\hat{P}_k(\mathbf{w_j})}{|| VS_k || + \mid VS_k \mid})^{-1}, \qquad (10)$$

where $|| \cdot ||$ returns the number of symbols and $\mid \cdot \mid$ the number of tokens in the set given as argument.

The iterative selection procedure runs as often as there are tokens in the largest class dependent sub corpus TS_k. One new token for each class Ω_k is selected in each iteration if the current's LM QF_k is lower or equal than the QF from the last iteration LQF_k. If so, the current token is marked as selected and no longer tested in the next iterations; the selection procedure continues with the following class. If the quality decreases, the next token is tested using the same procedure. If there are no possible tokens left for a class the LM trained with the tokens up to the iteration before is used for further training. After all classes have finished the current iteration, once more all LM run in parallel and the class dependent quality factor QF_k is computed and stored to the accumulators LQF_k. The procedure stops if the number of iterations is equal the number of training token of the largest sub corpus TS_k or if for no class a token was found that improved the recognition results.

5. DATA

For our experiments we used data from the VERBMOBIL-corpus for two different tasks: language identification (LID) and DA classification. In VERBMOBIL, the whole dialog of two persons is seen as a sequence of DA, which means that DA are the basic units on the dialog level. The DA are defined according to their illocutionary force, e.g., ACCEPT, SUGGEST, REQUEST, and can be subcategorized as for their functional role or their propositional content, e.g., DATE or LOCATION, depending on the application. In the VERBMOBIL-I domain, 18 DA are defined on the illocutionary level with 42 sub categories [3]. Since in spontaneous speech many incomplete and incorrect syntactic structures occur, e.g., a lot of elliptic sentences or restarts, it is not easy to give a quantitative and qualitative definition of the term DA. In VERBMOBIL, criteria were defined for the manual segmentation of turns based on their textual representation and for the manual labeling of these segments with DA [4]. No prosodic information is used for labeling, in order to be able to label the dialogs without listening to them. Thus it was possible to reduce the labeling effort.

Our DA training set has 19795 tokens and the test set has 2540. The training of our LM is done using a category system with 884 hand made categories, where we put for example citynames, surnames, first names and numbers in a category of their own. Words that occur very often in the training corpus have their own category. Each word only appears in one category, even if it could match more than one category definition.

In the LID experiment, were we want to decide if an English utterance is spoken by an American native speaker or by a German, we used vector quantized data [2] from the English subset of the VERBMOBIL-corpus as symbol sequences. Our training set contains 920, the validation set 200 and the test set 100 tokens. Each training set contains 40 tokens of the other class to see if our method is able to eliminate these from the training set.

6. EXPERIMENTAL RESULTS

For our bootstrap training experiments we restrict our maximum polygram size to bigrams, because of the very high computation time for one experiment. Furthermore, we selected for all experiments the same 10% of the training data for the validation set randomly. Thus, the results are comparable. The rational interpolation method was used for all experiments. No optimization of the interpolation coefficients was done to observe the improvement produced by the bootstrap training. The results for the experiments with the PX_k and RR_k as QF of the bootstrap procedure are presented in Figure 3 for $\overline{RR}$ and in Figure 4 for RR. As can be seen in Figures 3 and 4, the approach with RR_k as QF

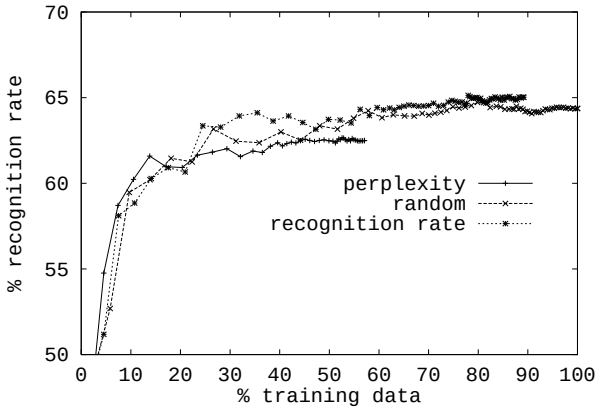

Figure 3: $\overline{RR}$ on an independent test set for DA classification after each iteration for random selection, PX_k and RR_k as QF.

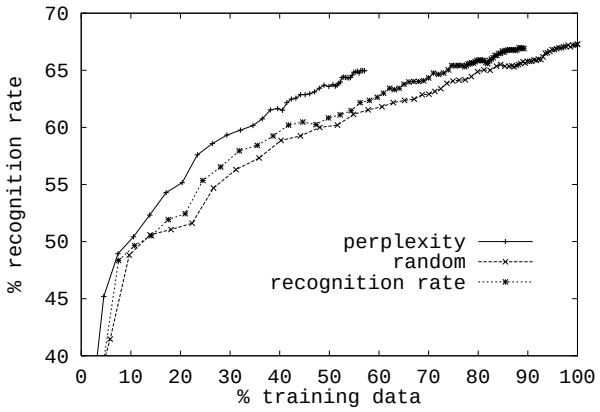

Figure 4: RR on an independent test set for DA classification after each iteration for random selection, PX_k and RR_k as QF.

obtains slightly better results for the $\overline{RR}$ and the same results for RR with only 88% of the training data. The perplexity curve is steeper in the early part of the experiment but could not reach the results from the RR_k approach. The random curve is our baseline system, which we use as motivation for our approach.

We ran the same experiments for the small LID set. In Figure 5 the RR curve for RR_k as QF is presented. One can see that the QF criterion (RR_k) works very well on this smaller two class problem. It is interesting to note that the best results on the independent test set were achieved with only 20% of the training data, even though almost all tokens are used because the RR on the VS does not decrease. All 17 tokens that were excluded from the training were these from the 40 German utterances in the English speakers set.

7. CONCLUSION AND FURTHER RESEARCH

Rosenfeld already said: the time-honored maxim "there's no data like more data" no longer holds; with our bootstrap approach we have shown that it is possible to train class dependent LM for DA classification and LID on subsets of training corpora with slightly better results and to exclude wrong training samples. Our future work is to advance this approach to get higher improvements and

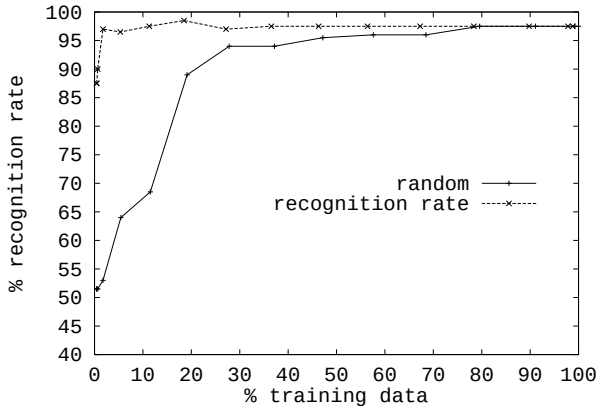

Figure 5: RR on an independent test set for the LID experiment after each iteration for random selection and RR_k as QF.

to run experiments in other domains like topic spotting. This can be done for example with other QF like precision and less restricted search spaces for the selection procedure. Since those approaches are very computationally expensive we first want to speed up our LM training procedure.

8. REFERENCES

1. T. Bub and J. Schwinn. Verbmobil: The Evolution of a Complex Large Speech-to-Speech Translation System. In *Int. Conf. on Spoken Language Processing*, volume 4, pages 1026–1029, Philadelphia, 1996.
2. S. Harbeck, E. Nöth, and H. Niemann. Multilingual Speech Recognition in the Context of Multilingual Information Retrieval Dialogues. In *Proc. of the Workshop on TEXT, SPEECH and DIALOG (TSD'98)*, Brno, 1998. Masaryk University. (to appear).
3. S. Jekat, A. Klein, E. Maier, I. Maleck, M. Mast, and J. Quantz. Dialogue Acts in Verbmobil. Verbmobil Report 65, 1995.
4. M. Mast, E. Maier, and B. Schmitz. Criteria for the Segmentation of Spoken Input into Individual Utterances. Verbmobil Report 97, 1995.
5. H. Ney, U. Essen, and R. Kneser. On Structuring Probabilistic Dependences on Stochastic Language Modelling. *Computer Speech & Language*, 8(1):1–38, 1994.
6. R. Rosenfeld. Optimizing Lexical and N-gram Coverage via Judicious Use of Linguistic Data. In *Proc. European Conf. on Speech Communication and Technology*, volume 3, pages 1763–1766, Madrid, 1995.
7. E. G. Schukat-Talamazzini. *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen*. Vieweg, Braunschweig, 1995.
8. E.G. Schukat-Talamazzini, F. Gallwitz, S. Harbeck, and V. Warnke. Rational Interpolation of Maximum Likelihood Predictors in Stochastic Language Modeling. In *Proc. European Conf. on Speech Communication and Technology*, volume 5, pages 2731–2734, Rhodes, Greece, 1997.
9. M. A. Zissmann and E. Singer. Automatic language idendification of telephone speech messages using phoneme recognition and n-gram modeling. pages I–305 – I–308, Adelaide, April 1994.