Gefördert vom
bmb+f
Bundesministerium für
Bildung, Wissenschaft,
Forschung und Technologie

Verbmobil
Verbundvorhaben

Topic Spotting Using Subword Units

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

F.–A.–Universität Erlangen–Nürnberg

Vm
Report 205
März 1997

März 1997

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

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

Tel.: (09131) 85 - 7883
e-mail: warnke@informatik.uni-erlangen.de

Gehört zum Antragsabschnitt: 4.8

Das diesem Bericht zugrundeliegende Forschungsvorhaben wurde mit Mitteln des Bundesministers für Bildung, Wissenschaft, Forschung und Technologie unter dem Förderkennzeichen 01 IV 701 K/5 gefördert. Die Verantwortung für den Inhalt dieser Arbeit liegt bei den Autoren.

From: Tagungsband des 9. Aachener Kolloquiums "Signaltheorie", pages 287 - 290

Topic Spotting Using Subword Units

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

Universität Erlangen–Nürnberg, Lehrstuhl für Mustererkennung,
Martensstr. 3,
91058 Erlangen, Germany
Tel.: +49 9131 / 857883
Fax.: +49 9131 / 303811
warnke@informatik.uni-erlangen.de http://www5.informatik.uni-erlangen.de

Abstract

In this paper we present a new approach for topic spotting based on subword units and feature vectors instead of words. In our first approach, we only use vector quantized feature vectors and polygram language models for topic representation. In the second approach, we use phonemes instead of the vector quantized feature vectors and model the topics again using polygram language models. We trained and tested the two methods on two different corpora. The first is a part of a media corpus which contains data from TV shows for three different topics. The second is the VERBMOBIL-corpus where we used 18 dialog acts as topics. Each corpus was splitted into disjunctive test and training sets. We achieved recognition rates up to 82% for the three topics of the media corpus and up to 64% using 18 dialog acts of the VERBMOBIL-corpus as topics.

1. Introduction

In most approaches in the field of topic spotting, words or word sequences are used for identifying a topic [5]. This is done by word recognition with large vocabularies or special word spotters [9]. To train such recognizers in both cases a huge amount of tediously labeled data has to be available.

For the training of our topic spotter with phonemes and vector quantized feature vectors we do not need the data to be labeled as exactly as for training a word spotter or a word recognition system. We only need the speech signals labeled with their topic rather than a word–to–word transliteration. Using either a vector quantizer (see section 2.1.) or phoneme segmentizer (see section 2.2.), we segment the speech signal into a symbol sequence. With these sequences we train stochastic language models (LMs) for each of the topics to be identified. In test phase we run all LMs in parallel and decide for the topic with the maximum a posteriori probability (see section 2.3.).

The advantage of our approach is evident when changing to a new domain. Doing topic spotting with a large vocabulary speech recognizer, one has to adapt the lexicon if not retrain all the acoustic models with domain dependent transliterated speech. With a word spotter, new keywords have to be identified and trained. In our approach only the language models have to be retrained. The training labels, i.e. the assignment of the topic to the training utterances, can be done very fast. Using the vector quantizer, one can even switch the language without the need for a more detailed labelling of the training data.

For our experiments we understand a topic in two different ways: in the first case a topic is a *theme of a TV show* (IDS-corpus) and in the second we understand a *dialog act* [2] as a topic (VERBMOBIL-corpus) (see section 3.).

2. Methods Used

2.1. Feature extraction and vector quantization

For representing the speech signal as feature vectors, we use the mel-frequency-cepstrum-coefficients (MFCCs). A feature vector $\mathbf{c}$ is calculated for a 10ms part of the speech signal and contains the energy and the first 11 MFCCs. These features and their first derivatives are used as an input to a phoneme segmentizer in the phoneme based approach. In the vector quantization approach eight neighboring feature vectors $\mathbf{c}$ (without the derivatives) are concatenated to a new feature vector $\hat{\mathbf{c}}$, which describes a context of 80ms with 96 coefficients. We use this time window, because the average length of a german phoneme is about 80ms. With this feature vector we calculate an initial codebook $q(\hat{C})$ with 240 classes. Using the *linear discriminant analysis* (LDA) on the codebook $q(\hat{C})$ we optimize the intra- and inter-class-distance of the 240 codebook classes and transform the feature vector $\hat{\mathbf{c}}$ with the 96 components in a new smaller vec-

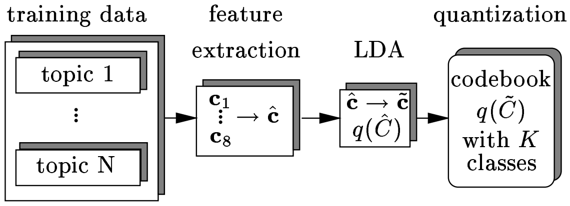

Figure 1: Partioning of the feature space with feature reduction using LDA

tor $\tilde{\mathbf{c}}$ with 24 components. This feature vector is used to train a new codebook $q(\tilde{C})$ with 65 classes (see figure 1). For training the codebook we use the well known *LBG*-algorithm [4], which minimizes the expected quantization error

$$\varepsilon = \mathcal{E}[d(\mathbf{C}, q(\mathbf{C}))] \tag{1}$$

With the resulting codebook we are able to segment the utterances of the training and test data into equidistant *codebook class sequences* (CCSs) with a segment length of 80ms. Using these CCSs we train a LM for every topic of interest.

2.2. Phoneme segmentation

In the phoneme based approach we use phoneme sequences instead of CCSs. Our phoneme segmentizer works HMM-based. Every phoneme is modeled by one *left-right-semicontinous HMM*. All HMMs are merged together to a *compound-HMM-network* with one start and one end state. Thus it is possible to segment an utterance and calculate the corresponding phonemes in one pass using the *one-pass*-algorithm as described in [3]. The inventory of our phoneme segmentizer is 62 phonemes and 3 silence models. We trained this phoneme segmentizer with the training data from the VERBMOBIL-corpus and used it to segment the data from the IDS-corpus as well.

We can see the phoneme segmentizer (just as the vector quantizer) as a signal–to–symbol transformation. The differences between the phoneme classes and the codebook classes is that codebook classes have no direct acoustic–phonetic meaning. Notice, that we keep the size of the input feature vector (24), the number of phonemes/codebook classes (65), and the average window size (80ms) in the same order. Thus we can see how much we loose in performance, when working with the unsupervised vector quantization rather than the supervised phoneme segmentation.

2.3. Polygram Language Model

In most cases language models are used to calculate the probability of a word sequence $\omega = \omega_1 \dots \omega_m$ in a given language or context. We use *polygram language models* [7], which are a special kind of *stochastic* language models to calculate the probability of a *symbol sequence* where a symbol could be a phoneme or a codebook class.

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

$$P(\omega_1 \dots \omega_m) = P(\omega_1) \cdot \prod_{n=2}^{m} P(\omega_n \mid \underbrace{\omega_1 \omega_2 \dots \omega_{n-2} \omega_{n-1}}_{\text{history}}) \tag{2}$$

Because the younger history $\omega_{m-n+1} \dots \omega_{m-1}$ of the symbol sequence $\omega_1 \dots \omega_m$ is more important for modeling and to restrict the number of free parameters inside the LM, we only use the last $n-1$ symbols instead of the whole history.

$$P(\omega_m \mid \omega_1 \dots \omega_{m-1}) \simeq P(\omega_m \mid \underbrace{\omega_{m-n+1} \dots \omega_{m-1}}_{(n-1)}) \tag{3}$$

With this shorter history we can estimate $P(\omega_m \mid \omega_{m-n+1} \dots \omega_{m-1})$ from a given training corpus using the interpolation scheme

$$\hat{P}(\omega_m \mid \omega_1 \dots \omega_{m-1}) = \frac{\#(\omega_1 \dots \omega_m)}{\#(\omega_1 \dots \omega_{m-1})}, \tag{4}$$

where # is a function which counts how often a symbol sequence is seen in the training data. To handle symbol sequences that were not seen in the training data we need an interpolation formalism.

Linear interpolation

The first interpolation method we use is the *linear interpolation* [7] (L = *lexicon size*):

$$\begin{aligned}\tilde{P}(\omega_m \mid \omega_1 \dots \omega_{m-1}) = p_0 \cdot \frac{1}{L} \\ +p_1 \cdot \hat{P}(\omega_m) \\ +p_2 \cdot \hat{P}(\omega_m \mid \omega_{m-1}) \\ + \sum_{i=3}^{n} p_i \cdot \hat{P}(\omega_m \mid \omega_{m-i+1} \dots \omega_{m-1}).\end{aligned} \tag{5}$$

The interpolation coefficients p_i can be estimated using the *Expectation Maximization (EM)* algorithm [6] on a given validation set. Using this method an unseen symbol sequence is modeled by its subsequences weighted with the interpolation coefficients.

Rational interpolation

The *rational interpolation* method is the second interpolation we apply [7]:

$$P(\omega_m \mid \omega_1 \dots \omega_{m-1}) =$$

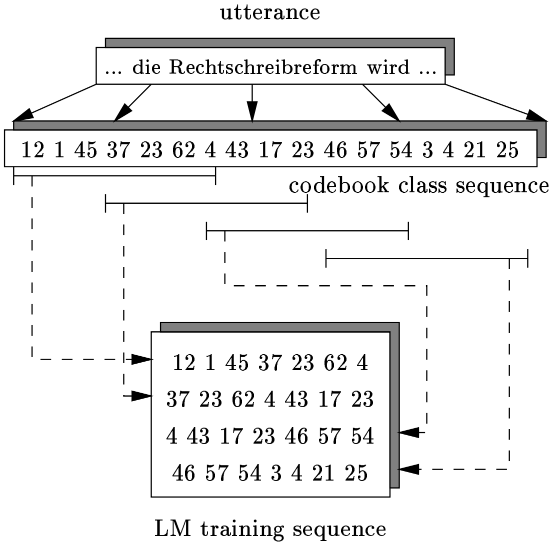

Figure 2: Splitting a CCS ω into subsequences with step $S = 3$ and window $W = 7$

$$\frac{\sum_{i=1}^{n} p_i \cdot (1/L)^{n-i} \cdot \#_i(\omega_1 \ldots \omega_{m-1}\omega_m)}{\sum_{i=1}^{n} p_i \cdot (1/L)^{n-i} \cdot \#_i(\omega_1 \ldots \omega_{m-1})}, \quad (6)$$

where $\#_i$ counts the i last predecessors $(\omega_{m-i} \ldots \omega_{m-1})$ in a given sequence ω. In this interpolation formalism it is also possible to estimate the interpolation coefficients using the EM-algorithm on a given validation set. This interpolation gives more weight to the symbol sequences which have often been present in the training data and are in the nearest neighborhood of the observed symbol.

3. Corpora

3.1. IDS-Corpus

We performed experiments for both approaches with the data from the IDS-corpus (Institut für Deutsche Sprache). This corpus contains data from German TV shows for the three topics *speech, politics, and culture*. It is a small part of the *IDS-media-corpus* with more than about 200 hours of spoken data. It was not easy to say, which TV show corresponds to which topic. So we assigned, for example, one TV show to the topic "politics", if it was announced that it is a political discussion about the "gulf war". All utterances of this TV show were then assigned to the topic "politics". 316 utterances from 11 different TV shows were divided into speaker disjunctive training (250 utterances) and test sets (66 utterances). The length of an utterance is 39 seconds in the average.

3.2. Verbmobil-Corpus

Verbmobil is a speech–to–speech translation project [1] in the domain of appointment scheduling, i.e. two persons try to fix a meeting date, time, and place. One of the tasks of the dialog module within Verbmobil is to

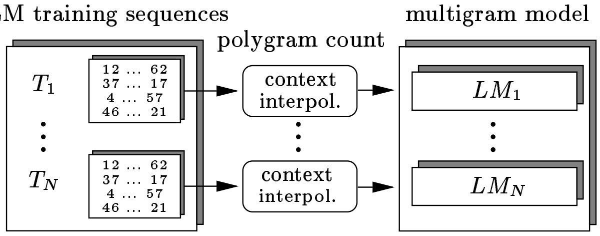

Figure 3: Training of LMs for the topics T_i

keep track of the state of the dialog in terms of dialog acts. Dialog acts are, e.g., "greeting", "confirmation of a date", "suggestion of a place". In Verbmobil a turn of one user can consist of more than one dialog act. Currently, the processing is done in two steps: first, the utterance is segmented into dialog act units. Second, these units are classified into dialog acts. In Verbmobil there are 19 dialog acts. In our experiments we recognize 18 of these dialog acts as topics[1]. We used 3284 dialog acts for training and 1064 dialog acts for testing. For the experiments described below we used a hand segmentation of turns into dialog acts.

4. Topic spotting

4.1. Preparing the Sequences

When we transform an utterance into a symbol sequence $\omega = \omega_1 \ldots \omega_m$ we can train and test our topic spotter with the whole symbol sequence or we can observe the sequence through a *sliding window* with window length W and *step* S (see figure 2) to get subsequences $\tilde{\omega}$. Thus we are able to get a result at each step of the recognition task and observe the provisional results.

4.2. Using Language Models

Using the sequence ω or $\tilde{\omega}$ we trained a polygram language model for each topic (see figure 3). For the classification task, we combine all the *polygram models* together to one *multigram model*. This multigram model contains the polygram models for all topics. Running the multigram model means running all polygram models in parallel to calculate the a posteriori probabilities $P(T_i \mid \tilde{\omega})$ for each topic $T_i \in \mathcal{T}$ of interest

$$P(T_i \mid \tilde{\omega}) = \frac{P(\tilde{\omega} \mid T_i) \cdot P(T_i)}{\sum_{j=1}^{N} P(\tilde{\omega}, T_j)}. \quad (7)$$

We decide for the topic with the maximum a posteriori probability. For the whole sequence ω (if we obsere $\tilde{\omega}$) we accumulate the hits for each subsequence $\tilde{\omega}$ and decide for the topic which receives the most hits.

[1] One dialog act was only present a few times in the training set and could thus not be trained.

5. Experiments and Results

Using the data from the IDS-corpus we calculated CCSs and used different length W for our sliding window. The step was in all experiments set to $S = 1$. So we were able to get a result for every 80ms of the utterance with the actual context of the window W. Table 1 shows the best results we achieved for different sizes of W and interpolation methods using polygrams of length three. We performed the same experiments with the

codebook class sequences with trigram - LM								
window	*10*		*20*		*30*		*whole*	
interpol.	R	L	R	L	R	L	R	L
speech	56	47	61	56	65	56	78	70
politics	74	77	70	81	70	74	78	85
culture	68	68	56	75	56	63	50	87
RR.	67	65	64	71	65	65	71	80

Table 1: Recognition results in percent for CCSs (L =linear interpolation, R =rational interpolation)

phoneme sequences. Here the window size W (W phonemes) is only approximately constant over time. We again reached the best results when we used the maximum context of three. The results for this experiment are given in table 2. We compared the results for our auto-

phoneme sequences with trigram - LM								
window	*20*		*30*		*40*		*whole*	
interpol.	R	L	R	L	R	L	R	L
speech	74	43	78	52	86	57	87	70
politics	81	81	85	89	79	85	74	81
culture	81	100	81	94	81	88	75	81
RR.	79	73	82	77	82	76	79	77

Table 2: Recognition results in percent for phoneme sequences

matically segmentized symbol sequences with LMs trained on the transliterated word sequences (simulating 100% word recognition) which reached a recognition result of 89% for a sliding window of $W = 10$ and 85% for the whole utterance. Thus we can see that is possible to classify the three topics of the IDS-corpus using subword units and polygram language models.

Using the VERBMOBIL-corpus as database we only did experiments with the phoneme sequences. At this point of our research we only used the spoken phoneme sequences of the 18 dialog acts for training and testing. We calculated the a posteriori probability for the whole symbol sequences of an utterance. Thus we reached a recognition rate of 64% for the 18 dialog acts when we used the phoneme sequences, a rational interpolation method and maximum polygram size of four for the language models. We also trained a topic spotter on the transliterated word sequence as we did for the IDS-corpus. Here we reached a recognition rate of 63%. This shows that (for the given training data) we can model the topics better with phoneme sequences than with the word sequences. One reason is that, e.g. the words *appointment* and *appointments* are different for the word based LMs, but the stem (appointment) is equal for the phoneme based LMs. A more detailed discussion of the experiments is given in [8].

6. Conclusion

We presented two approaches for topic spotting with the use of subword units. Because we use vector quantization and a domain independent phoneme segmentizer, we only need the utterances of the training set labeled with their topics. The main difference between both approaches is the fact, that we use CCSs of equidistant length in our vector quantizer and a symbol sequence of variable length in our phoneme based approach. In both approaches the topics are modeled with the help of stochastic language models. During the recognition task for all topics the a posteriori probability is calculated in parallel and the topic with the maximum probability is chosen. We showed that depending on the definition of topic and the amount of training data our approach performed almost as good or better than spotter which uses a “perfect” word recognition module.

7. 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. Jekat, A. Klein, E. Maier, I. Maleck, M. Mast, and J. Quantz. Dialogue Acts in Verbmobil. Verbmobil Report 65, 1995.
3. T. Kuhn. *Die Erkennungsphase in einem Dialogsystem*, volume 80 of *Dissertationen zur Künstlichen Intelligenz*. Infix, St. Augustin, 1995.
4. Y. Linde, A. Buzo, and R.M. Gray. An Algorithm for Vector Quantizer Design. *IEEE Trans. on Communications*, 28(1):84–95, 1980.
5. E. Parris and M. Carrey. Topic spotting with task independent Models. In *Proc. European Conf. on Speech Communication and Technology*, pages 2133–2136, Madrid, Spain, September 1995.
6. E. G. Schukat-Talamazzini. *Automatische Spracherkennung – Grundlagen, statistische Modelle und effiziente Algorithmen.* Vieweg, Braunschweig, 1995.
7. E.G. Schukat-Talamazzini. Stochastic Language Models. In *Electrotechnical and Computer Science Conference*, Portorož, Slovenia, 1995.
8. V. Warnke. Topik– und Topikgrenzen–Spotting. Diplomarbeit, Lehrstuhl für Mustererkennung (Informatik 5), Universität Erlangen–Nürnberg, 1996.
9. M. Wintraub. Keyword-spotting using sri's decipher large vocabulary speech recognition system. In *Proceedings International Conference on Automatic Speech and Signal Processing*, pages 463–466, 1993.