Automatic phonetic baseforms for person names in the Czech dialogue system

Pavel Brodský, Luděk Müller
University of West Bohemia in Pilsen, Department of Cybernetics,
Univerzitní 22, 306 14 Pilsen, Czech Republic
pbrodsky@kky.zcu.cz, muller@kky.zcu.cz

ABSTRACT

Phonetic baseforms are the basic recognition units in most speech recognition systems. These baseforms are usually determined by linguists once a vocabulary is chosen and not modified thereafter. However, several applications, such as name dialling require the user to be able to add new words to the vocabulary. These new words are often names or task-specific jargons that have user dependent pronunciation.

In this paper, we describe a method how to generate a phonetic baseform from acoustic pronunciation of a name without a prior knowledge of the name spelling. We used a language model based on bigram statistics.

1. INTRODUCTION

There has been considerable interest in *telecommunications and embedded speech recognition* application that provide personalized vocabularies. Name dialling is one such example of a telephonic application where it is necessary to have ability to provide speaker dependent vocabularies for repertory dialling. This feature enables the user to add even such words to the personalized vocabulary for which a spelling or acoustic representation does not exist in the speech recognition lexicon, and associate these words to a phone number to be dialled. We will show how speaker dependent baseforms could be derived from one or two speech utterances by using speaker independent acoustic model and a language model. We use the bigram probabilities to constrain the transition between phonemes.

The structure of this paper is as follows. In Section 2 we present our recognition system and its components: Speech recognition engine, acoustic modelling, front-end, labeller and decoder. In Section 3 we describe baseform generating algorithm. The mumble model is described in Section 4. In Section 5 language model for names and surnames of Czech Republic inhabitants is described. Experimental results are contained in Section 6 and Section 7 is conclusion.

2. SYSTEM OVERVIEW

The speech recognition engine is based on a statistical approach. It comprises a front-end, an acoustic model, a language model and a decoding block [1].

Acoustic Modelling: As a basic speech unit of the recognition system a triphone is used. Each triphone is represented by a 3-state left-to-right HMM with a continuous output probability density function assigned to each state. Each density is expressed as a mixture of multivariate Gaussians with a diagonal covariance matrix. The Czech phonetic decision trees were used to tie states of Czech triphones.

Front-end: The speech signal is digitized through a telephone board at 8 kHz sample rate and converted to the mu-law 8-bit resolution format. The parametrization process used in our system is as follows: Firstly the pre-emphasized acoustic waveform is segmented into 25 millisecond frames every 10ms. A Hamming window is applied to each frame and 13 MFCCs (including the energy coefficient c_0) are computed. The first-order and second-order derivates of MFCCs are computed and appended to the static MFCCs each speech frame.

Labeller: The recognition algorithm uses 2510 different tie states, each of which represented by a mixture of 8 Gaussian distributions in the 39-dimensional space. Thus during a decoding it is necessary to compute a large number of log-likelihood scores (LLSs) every 10ms. In order to perform the recognition in real time the number of calculations is reduced by applying a technique which seeks to find and precisely determinate only first 150 most probable LLSs. This technique efficiently uses relevant statistical properties of the Gaussian mixture densities combining them with "a priority hit" technique and the kNN method. This approach allows more than 90% reduction of a computation cost without substantial decrease recognition accuracy.

Decoder: The decoder uses a crossword context dependent HMM state network generated by a Net generator. The input of the Net generator is a text grammar format represented by an extended BNF with respect of JSGF. The whole net consists of one or more in run-time connected regular grammars. A considerable part of the net is usually generated before the decoder starts but every part of the net can be generated on

© UkrObraz'2002

demand in run-time. The decoder utilizes a Viterbi search with a beam pruning.

3. PHONETIC BASEFORMS

Phonetic baseforms are the basic recognition units in most speech recognition systems. These baseforms are usually determined by linguists once a vocabulary is chosen and not modified thereafter. However, several applications, such as name dialling, require the user be able to add new words to the vocabulary. These new words are often names or task-specific jargons that have user dependent pronunciations.

The phonetic baseform is sequence of phonemes which represents a given utterance (name). We can create it manually (by phonetic transcription rules) or automatically.

Example of baseform (for the Czech phonetic alphabet):

Name:	Luděk Müller
Manually:	sil l u dj e k sil m i l e r sil
Automatic:	sil r i d e p t sil m e l aa r sil

Some utterances can be pronounced by several ways depending on speaker, speaking style, and other conditions. Therefore, generally more than one baseform can be constructed and stored for an utterance (e.g. a person name). In this work we used only one phonetic baseform for each person name.

Each baseform can be easily added manually. In the case of automatic baseform generation the user for example can be required to say the given person names twice and for each utterance a baseform should be automatically generated.

The baseform generation algorithm is based on Viterbi algorithm [3] that searches the best phoneme path through the HMM net consist of all Czech phonemes and a set of phoneme transitions. The net structure is dependent on the language (phoneme) model and can be interpreted also as so-called mumble model described in more detail in the Section 4.

Also N-best hypotheses instead the first best hypothesis can be considered and in this case the decoder should produce a list of the most probable phoneme sequences. The recognition results can be stored as well in a phoneme graph which is an analogy to the word graph in the case of N-best word sequences decoding problem.

4. MUMBLE MODEL

The mumble model is constructed as a set of HMM [4] models connected in a parallel fashion. Each HMM model is 3-state left-to-right and represents one context-independent phone. The structure of the mumble model is depicted in Figure 1. Actually the probabilities of emission of an observation vector in a given state are evaluated as the maximal emission probability of all corresponding states of context-dependent triphones. Thus neither additional HMM models nor additional training is required. The value of the backward loop probability *BPr* causes a various length of the phone sequence recognized by the network in Figure 1. While the higher value produces more insertions, the small value induces more deletions in the resulting phone sequence.

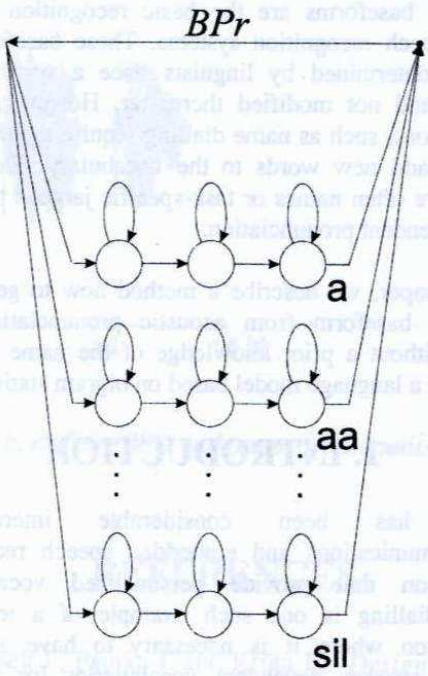

Figure 1.

In Figure 2. is mumble model with language model basis on bigrams with full matrix of transition probabilities. Language model is described in Section 5.

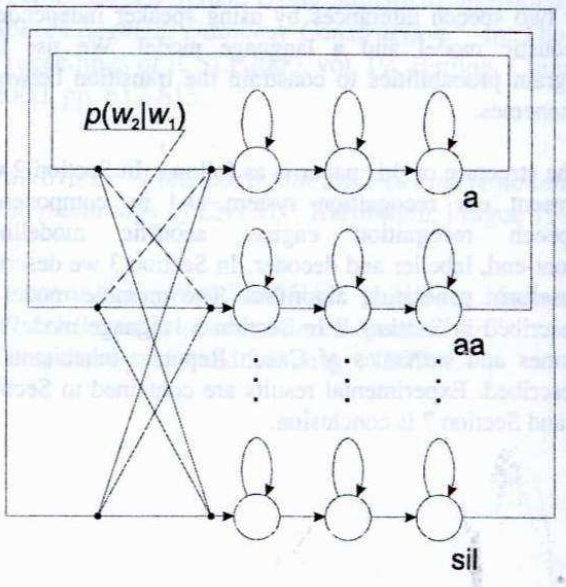

Figure 2.

5. LANGUAGE MODEL

The language model generally restricts variety of words sequences W and consequently also phone sequences. This restriction may be either deterministic (i. e. some words or phones sequence are not allowable) or "softer" stochastic (some word sequence are less probable). In our case we chose a probabilistic approach and an absolute discounting language model with backing-off for conditional probabilities method.

The basic idea of the absolute discounting language model with backing-off for conditional probabilities is to keep a high number of joined events (h, w), a word history h and a word w, almost unmodified. We suppose that the number of occurrences of joined events in training text will not change probably too much, if we select another training text of the similar size (from the same problem area). To consider possible variability of the number $N(h, w)$ of occurrences (h, w) in text we introduce parameter of permanent deviation b_h, so-called absolute discounting parameter that decreases the number of seen events $N(h, w)$. Furthermore we suppose that b_h is not dependent directly on the value $N(h, w)$, nevertheless it is dependent on the history h. This deviation must remain negative because unseen events in the text requires nonzero (thus positive) probabilities. By means of absolute discounting parameter b_h the part of probability mass is redistributed from the seen events to unseen events. The resulted formulae for the absolute discounting language model is:

$$\bar{p}(w|h) = \begin{cases} \dfrac{N(h,w) - b_h}{N(h,.)}, & \text{for } N(h,w) > 0 \\ b_h \dfrac{V - n_0(h,.)}{N(h,.)} \cdot \dfrac{\beta(w|\bar{h})}{\sum_{w':N(h,w')=0} \beta(w'|\bar{h})}, & \text{for } N(h,w) = 0 \end{cases} \quad (1)$$

where $\beta(w|\bar{h})$ is a conditional probability of observing the word w given the generalized history $\bar{h}$. The generalized (also reduced) history $\bar{h}$ is defined as:

If (h,w) n-gram $(w_1, w_2, ..., w_n)$, then $(\bar{h}, w)$ is $(n\text{-}1)$-gram $(w_2, ..., w_n)$.

Our language model was created for names and surnames occurring in Czech Republic. We had at disposal 4 137 different names and 236 769 different surnames. In total we had 10 282 470 names and 10 296 459 surnames. As a basic unit of the language model we chose a phoneme which means that before computing individual probabilities we had to perform a phonetic transcription. Foreign names and surnames that are not subjected to Czech phonetic transcriptions rules were transcribed manually and saved into vocabulary of exceptions. After the phonetic transcription we computed probability of unigrams, bigrams and trigrams according to equation (1). Results are shown in Table 1. including the task perplexity and entropy.

Characteristics of training and test corpus			
		training	test
number of names		20 579 070	9 488
number of phonemes in vocabulary (V)		44	44
unigrams	unigrams (N)	174 692 140	77 100
	different unigrams (nr(.,.))	44	43
	unseen unigrams (n0(.,.))	0	1
	singletons (n1(.,.))	0	0
	doubletons (n2(.,.))	0	0
	perplexity	19.22	19.03
	entropy	4.26	4.25
bigrams	bigrams (N)	154 112 095	67 612
	different bigrams (nr(.,.))	1 392	852
	unseen bigrams (n0(.,.))	544	1 084
	singletons (n1(.,.))	35	101
	doubletons (n2(.,.))	13	60
	perplexity	9.95	10.11
	entropy	3.32	3.34
trigrams	trigrams (N)	133 532 050	58 124
	different trigrams (nr(.,.))	18 904	4 999
	unseen trigrams (n0(.,.))	66 280	80 185
	singletons (n1(.,.))	1 198	1 725
	doubletons (n2(.,.))	645	757
	perplexity	4.70	4.54
	entropy	2.23	2.18

Table 1.

6. EXPERIMENTAL RESULT

In the speech recognition system equipped by an acoustic and a language model it is practically advantageous to use different weights of the language and the acoustic model. These weights can be defined by two variables p a s. The word insertion penalty p is a fixed value added to each token when it transits from the end of one word to the start of the next. The grammar scale factor s is the amount by which the language model probability is scaled before being added to each hypothesis as it transits from the end of the word to the start of the next. These parameters can have a significant effect on recognition performance and hence, some tuning on development test data is well worthwhile.
Formulae for computing with parameters s and p:

$$\log(P(O, w_k|w_l)) = \log(P(O|w_k)) + s \cdot \log(P(w_k|w_l)) + p \quad (2)$$

where $\boldsymbol{O}$ is observation vector sequence generated by Hidden Markov Model; w_k and w_l are phonemes and P is a likelihood.

After the language model had been created we performed several tests. The first test was performed using 718 utterances. Each utterance consists of person name and surname. From 718 utterances two sets A, B were randomly chosen. Each of them contained 25 different utterances. Two training sets were constructed. The first one (T1) is equal to the set A and the second one (T2) is composes both sets A and B. The test set contained all 718 utterances. From sets A and B the baseforms were generated. The test set was recognized on basis these baseforms. Results are shown in Table 2. for the grammar scale factor $s = 1$ and the word insertion penalty $p = 30\,000$.

Number of baseforms	25 (T1)	2x25 (T2)
Number of utterances	718	718
Correctly recognized	554	601
Incorrectly recognized	164	117
Correctly recognized [%]	77.16	83.70
Incorrectly recognized [%]	22.84	16.30

Table 2.

Following series of tests were performed with the same set of 718 utterances. 25 utterances were always randomly chosen (one for every name) and all the 718 utterances were subsequently recognized with various values s and p. General results shown in Table 3. are arithmetic mean of all individual tests.

s/p	20000	25000	30000	35000	40000	50000
-1	75.81	75.07	72.28	68.71	64.02	59.52
0	79.53	79.20	75.91	74.74	70.98	63.37
1	79.02	78.46	77.21	76.42	74.47	68.48
2	76.04	78.55	79.06	77.11	76.93	74.65

Table 3.

In the final test we tried how the system works for one user. We recorded 200 utterances by one user (4 utterances for each full name from the test A. From 200 utterances we randomly chose 25 utterances of different names and for each utterance a baseform was generated. The rest (175) utterances were recognized with these baseforms. We executed two tests with different training utterances. Results shown in Table 4. are arithmetic mean of both the tests.

s/p	20000	25000	30000	35000	40000	50000
0	94	93	92	90,5	88	84
1	93.5	92.5	93	95.5	91.5	87.5
2	99.5	97.5	91.5	89	89	90.5

Table 4.

7. CONCLUSION

We have presented a system for recognition and automatic phonetic baseform generation. We used an acoustic model and a language model with bigram probabilities to constrain the transitions between phonemes.

In conclusion, we believe that we have a viable technique for automatic generation of phonetic baseforms that give a good decoding accuracy with our speech recognition system. This is particularly useful for our telephony dialogue system where personalized vocabularies are a must.

Experimental results show that the recognition based on acoustic baseforms has a good accuracy. The highest achieved accuracy ($s = 2$, $p = 20\,000$) for system working with one user is greater then 99 %.

In the future we want to provide further improvements in accuracy.

REFERENCES

[1] Müller L., Psutka J., and Šmídl L. "*Design of Speech Recognition Engine*", TSD 2000 – Third International Workshop on TEXT, SPEECH and DIALOGUE, Brno, Czech republic, September 13-16, 2000.

[2] Ramabhadran B., Bahl L. R., deSouza P. V., Padmanabhan M., "*Acoustic-Only Based Automatic Phonetic Baseform Generation*", ICASSP 1998

[3] Forney, G. D., Jr. „*The Viterbi Algorithm*", In Proceedings of the IEEE, vol.61, no.3, pp.268-278 1973

[4] Young, S., Evermann, G., Odell, J., Ollason, D., Woodland, P., Kershaw, D., Moore, G., Valtchev, V. „*The HTK Book (for HTK Version 3.1)*", Cambridge University 2001