DEVELOPMENT OF A SPEECH RECOGNIZER FOR THE DUTCH LANGUAGE

Pascal Wiggers, Jacek Wojdel, Leon Rothkrantz
Data and Knowledge engineering,
Delft University of Technology,
Mekelweg 4, 2628 CD Delft, The Netherlands
Email: P.Wiggers@its.tudelft.nl,
J.C.Wojdel@its.tudelft.nl,
L.J.M.Rothkrantz@its.tudelft.nl

KEYWORDS
Speech recognition, Hidden Markov Models

ABSTRACT

This paper describes the development of a large vocabulary speaker independent speech recognizer for the Dutch language. The recognizer was build using Hidden Markov Toolkit and the Polyphone database of recorded Dutch speech. A number of systems have been build ranging from a simple monophone recognizer to a sophisticated system that uses backed-off triphones. The system has been tested using audio from different acoustic environments to test its robustness. The design and the test results will be presented.

INTRODUCTION

This paper describes the design and construction of a speech recognizer for the Dutch language. The system was created as a baseline system for further research on the subject of speech recognition within our group; in particular on the integration of information from multiple modalities in a Hidden Markov based speech recognizer in order to create a natural and robust human computer interface. Although a number of well-performing speech recognizers is now available, we decided to create our own system to ensure that it fits our needs and that it can easily be altered and extended.

The system was designed to recognize sounds recorded by an ordinary desktop computer microphone. To make the system as general as possible it was decided to create a speaker independent large vocabulary continuous speech recognizer and to make the system easy adaptable to new vocabularies it was decided to create a phoneme based recognizer.

The final system uses context dependent models, but as this system was build an refined incrementally, actually a whole set of recognizers has been created ranging form a simple monophone recognizer to a sophisticated multiple mixture triphone system.

This paper first describes the tools and data that were used to create the system it then gives a brief description of the development process and in the last section experimental results concerning the performance of the system are presented.

DEVELOPMENT OF THE RECOGNIZER

The system was developed using a variety of tools, programs and scripts. By far the largest set of tools and software libraries was taken from the Hidden Markov Toolkit (HTK). This is a portable software toolkit for building and manipulating systems that use continuous density Hidden Markov Models. It has been developed by the Speech Group at Cambridge University Engineering Department (Young et al. 1995). HTK provides the means to create and manipulate Hidden Markov models in general but it is primarily designed for building HMM based speech-processing tools.

The toolkit comprises a number of script-driven tools supported by a set of software libraries. Furthermore a number of programs was written that filled the gaps in the development process not covered by any of the tools. This includes a program for data selection, some tools for creating an initial acoustic model set and a tool for creating scripts for triphone clustering.

The system was built in four stages. First data for training and testing was selected and prepared. Then a simple monophone system was created. This system was subsequently refined and finally a number of evaluation tests were conducted to measure the performance of the system. Each of these steps will be described in the following sections.

Data preparation

The training and testing data for this project was taken from the Dutch Polyphone database (Damhuis et al 1994; Boogaart et al. 1994). This is a rather large corpus containing telephone speech from 5050 different speakers in 222075 speech files, based on 44 or in some cases 43 items per speaker. The speakers were selected from all dialect regions in the Netherlands and the ratio between male and female speakers is almost fifty-fifty. The utterances contain all Dutch phonemes in as many phonetic contexts as the designers of the database could find.

As the Polyphone database was recorded with automatic voice-interactive telephone services in mind most speech files contain examples of phrases useful for this kind of applications, this includes street names, bank-accounts, numbers and answers to yes-no questions. Training a large vocabulary recognizer on these samples may result in a recognizer that performs well on recognizing numbers and

'yes' and 'no' but which generalizes very poor to other words. To avoid these problems only nine items per person were used. All sentences were selected from newspaper articles. Five of these sentences belonged to the group of phonetically rich sentences that were selected to cover as many phonetic contexts as possible. The other four sentences were selected because they contained common frequently used application words.

The telephone recordings contained many samples of poor quality or contained background noise or even background speech. To ensure well-trained models that can be used for recognizing speech recorded for example using a PC microphone only utterances that were spoken by native speakers and which did not contain any background speech, background noise, no stuttering or disturbing hesitations and no mispronunciations or foreign pronunciations were selected. Mouth noises, like smacking, sniffing, loud breaths and verbal hesitations however were allowed.

Selection of the utterances that adhered to this profile was done automatically using meta-information provided by the Polyphone database. For each selected utterance a word level transcription was created.

Three different data sets were extracted from the Polyphone database this way. A training set containing 22626 utterances, a development set containing 2673 utterances, which was used for testing and fine tuning during development of the system and an evaluation test set comprising 2885 utterances, to evaluate the final performance of the system. The development set contained persons and sentences that did not occur in the training set. And the evaluation test set contained persons that did neither occur in the training set nor in the development set, but its phonetically rich sentences did also occur in the training set.

Bigram language models for testing and evaluation were calculated using the training data. The models were smoothed to incorporate words that did not occur in the training data using a backing-off scheme. Part of the probability mass from other words was transferred to these words.

To test how well the final system would adapt to other environments we recorded a small data set using a digital video camera (Wojdel, 2001). This set contains data from 5 different persons, all of them computer science students at TU Delft, four male students and one female student. From each person four or five recording sessions were used. Each session contained 23 sentences, ten of which were phonetically rich sentences, similar to those in the Polyphone database. The other sentences contained either a sequence of short words, a sequence of numbers, a spelled word or a command from a telebanking application.

Since the Polyphone utterances were encoded in 8-bit A-law wave format we converted our recordings, that were originally in 16-bit 44 kHz stereo wave format, to A-law format. Subsequently, all utterances were encoded to Mel-frequency cepstral coefficient vectors. Each vector contains twelve cepstral coefficients with log energy and delta and acceleration coefficients added, all scaled around zero by subtracting the cepstral mean from all vectors. This resulted in 39 dimensional feature vectors. A sampling rate of 10 ms was used and each vector was calculated over a segment of 25 ms.

The phonemes from the SAMPA set (Boogaart et al. 1994) were adopted as phoneme set for the recognizer. Three special purpose phonemes were added. The first, *sil*, models (longer periods of) silence that occur between sentences or when a person is not speaking at all. The second phoneme, *sp*, also represents silence, but only optional periods of short duration, like the silences that occur between words. The third phoneme that was added *mn*, models all kinds of mouth noise and verbal hesitations. The idea behind the inclusion of this phone was that real, natural speech always contains mouth noise and modeling this may improve the results in real-life environments. Verbal hesitations like ‘uh’ and ‘ehm’ were modeled by including them in the dictionary like any other word. The resulting set contained 45 phones.

For each of these phones a Hidden Markov Model was created. All models except the silence models shared the same topology, which consisted of non-emitting start and end states and three emitting states using single Gaussian density functions. The states are connected in a left-to-right way, with no skip transitions. Each state has a transition to itself. The model is shown in figure 1.

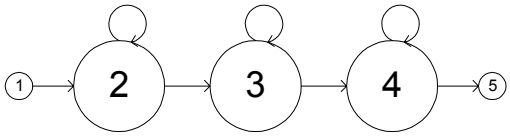

Figure 1: Acoustic HMM Topology

This is a rather simple model topology, but earlier experiments with three and five state models showed that the three state model performs as well, and in some cases even slightly better than the five state model, despite the fact that it has fewer parameters (Wiggers, 2001).

Training

As only unsegmented training data was available we initially set the mean and variance of all the Gaussians of all the models to the global mean and variance of the complete data set. These models were then trained using embedded Baum-Welch re-estimation. In the embedded re-estimation algorithm for each utterance the transcription is used to create a composite HMM which spans the whole utterance by concatenating instances of the phone HMMs corresponding to each label in its transcription. The Forward-Backward algorithm is then applied. When all of

the training files have been processed, the new parameter estimates are formed and the updated HMM set is created. This way no boundary information for the acoustic models is needed.

To make the system robust the *sp* silence model was added only after the models received some training. This was done because the pause model, which models optional short periods of silence, is very susceptible to picking up vectors belonging to neighboring phones. So in this particular case it is beneficial to explicitly define what type of data this model is supposed to represent. This was realized by creating a one state *sp* model, which has a direct transition from entry to exit node. It was then tied to the middle state of the silence model, so these two states now shared the same set of parameters. To the silence model *sil* transitions were added from the second state to the fourth state and back from the fourth state to the second state to make sure that the model could handle great variations in durations, from milliseconds up to a few seconds. This also had to be done after initial training to prevent that the silence model absorbed large parts of the utterances.

After re-estimation the models created so far were used to create new transcriptions using pronunciations that fitted the acoustic data embodied in the models. This was done by performing Viterbi alignment. These new transcriptions were then used in subsequent re-estimation cycles.

The performance of the single Gaussian monophone system that was created this way was tested by using the Viterbi decoding algorithm on the development test set. The transcriptions output by the Viterbi algorithm were compared to the original word level transcription files by a analysis tool, that uses a dynamic programming based string alignment procedure that is fully compatible with the one used in the standard US NIST scoring package. The percentage of word that was recognized correct was calculated, as was the so-called word accuracy. This measure compensates for the fact that the scoring algorithm may align incorrect inserted words with correct words by subtracting the number of insertion errors from the number of recognized words before calculating percentages.

In this test and in all other tests conducted during development that are described below, the first half of the development test set was used. This subset comprises 240 utterances spoken by about 100 different persons. The subset contained 1060 different words. A bigram language model containing these words was used in these tests. Table 1 shows the recognition results from the monophone system. To show the progress that has been made during training the results of various steps are included. Although the improvements made were considerable the overall results were very poor. This is due to the simple model topology and the fact that the system does not take into account linguistic effects like coarticulation and it does not make up for possible unbalances in the training data.

Table 1: Results Monophone System

System	Percentage of words recognized	Word accuracy percentage
Initial model set	17.15%	-77.17%
Fixed silence models	30.00%	-48.75
Monophone system	38.34%	-28.42%

To find more realistic density functions for the acoustic models Gaussian mixture densities were used. These were obtained from the simple Gaussian densities by iteratively incrementing the number of mixtures. For the distribution in a state the mixture with the largest weight was split until the required number of components was obtained. To prevent defunct mixtures with zero weight a floor mixture weight was defined, mixture weights were not allowed to fall below this floor. It turned out that during training the number of floored mixtures increased rapidly. To reduce this number and prevent overfitting the data the minimum number of examples necessary to allow for mixture incrementing of a model was also incremented in each step. The mixtures were incremented in seven steps until a 15-mixture system was obtained. The recognition results for these systems on the development test set are shown in table 2.

Table 2: Multiple Mixture Systems

Number of mixtures	Percentage of words recognized	Word accuracy percentage
2	40.11%	-25.42%
3	42.70%	-16.33%
5	45.74%	-7.57%
7	47.92%	-1.81%
10	51.54%	4.77%
12	54.50%	9.54%
15	56.81%	15.51%

To capture coarticulation effects context dependent models were introduced. In this project it was decided to use word internal triphones (which implies the use of biphones at word boundaries). These were created by cloning the monophone models as often as needed and creating triphone transcriptions to re-estimate the new model set. This way a set containing 8570 triphones was obtained. For many of these models there only was a single example in the training data, so the models could not reliably be estimated.

To find the right balance between the number of models and their modeling accuracy data-driven state clustering was used to obtain a smaller set of generalized triphones. A weighted Euclidean distance between the means of the Gaussian distribution functions was used to create clusters for each of the three model states. 15 different clusterings

were tried using different minimum cluster sizes to find the right number of triphones. Out of these clusterings five different systems were build and re-estimated. For each system the transition matrices of the triphones that corresponded to the same monophone were tied. Models that ended up having their states in the same clusters and which had the same transition matrix were tied. Thus in this case one physical model represents several logical triphones.

The systems were tested using the development test set and the corresponding bigram. As is clear from table 3 the recognition results improved as more triphones were used.

Table 3: Triphone Systems

Number of triphones	Percentage of words recognized	Word accuracy percentage
101	40.44%	-25.46%
563	46.48%	-10.28%
1050	49.57%	-4.77%
2526	54.92%	5.68%
8570	62.32%	18.59%

Unfortunately this also means a larger model set. The system containing 2526 triphones booked fairly reasonable results. It had at least three models per cluster and in most cases more. Each cluster had at least four examples in the training data. This system seemed to provide a good balance between the number of parameters and the modeling accuracy. It contained less than one third of the original triphones, but was still large enough to model different contexts, therefore it was chosen to be further developed during subsequent steps.

However, before these steps could be performed one problem had to be solved. A limitation of the data-driven clustering technique is that it does not deal with triphones for which there are no examples in the training data. This may be avoided by careful design of the training database but a little research showed that this was not an option in this case. The training data contained 8570 triphones out of 10205 in the dictionary that was used. Redistributing the available data among the data set would not have solved the problem as the evaluation data set and the test data set together only contained 394 additional triphones. Furthermore there was no guarantee that our dictionary contained all possible triphones.

A well known solution to these problems is the use of decision tree based clustering (Woodland et al., 1994a; Jelinek, 1999), however, this is a knowledge based approach, using phonetic knowledge to classify triphones and we were interested in using a completely data driven approach. Therefore we introduced a technique that can be described as 'backed-off triphone approach'. In this approach the original monophone models augment the triphone model set. During recognition the word network is constructed by inserting the HMMs in the language model, triphone models are used whenever available, otherwise the corresponding monophone is used. This is implemented by tying all triphones that have no model of their own to the corresponding monophone. Essentially the monophones become generalized triphones. Of course they are less specialized than the other generalized triphones because they are trained on all corresponding triphones but the ones they represent, but being monophones they are general enough to cover the unseen triphones. The overall result is a robust recognizer that uses triphones most of the time and does not break down when an unknown triphone is encountered

The triphone models still had Gaussian distribution functions. As with the monophone system the mixtures were incremented in steps of two or three mixtures a time, with subsequent reestimation cycles. This process was stopped at 17 mixtures because the relative gain in performance introduced by additional mixtures got to small. A 17-mixture system performed almost as well as a 19-mixture system. The recognition results of these systems are shown in table 4.

Table 4: Multiple Mixture Triphone Systems

Number of mixtures	Percentage of words recognized	Word accuracy percentage
2	56.27%	8.02%
3	58.82%	15.63%
5	61.00%	22.21%
7	64.05%	26.82%
10	66.64%	30.93%
12	68.61%	34.92%
15	70.55%	38.46%
17	71.00%	39.57%

EVALUATION EXPERIMENTS

The final set of acoustic models consisted of the 17-mixture generalized triphone set and the 15-mixture monophone set. This system was tuned using the development test set. A grammar scale factor was used to regulate the relative influences of the language model and the acoustic model. The optimal system, which relied mainly on its acoustic models, had a word recognition percentage of 95.27%, a word accuracy of 89.59% and 38.43% of all sentences were recognized correctly.

These results were obtained on the test set that was used during the development process to tune several parameters therefore they are likely to be too optimistic. To test the robustness of the system and to see how it will perform on other data we did a number of evaluation tests.

First recognition was performed on part of the evaluation test set. This set contained 100 sentences, each of which was spoken by a different person. The bigram language model used contained 5017 different words. The result of this and subsequent tests is shown in table 5. The

experiment showed that the recognizer generalizes very well to speakers it was neither trained nor tuned on even when a large word network is used. Although the evaluation data did not occur in the training or development test set it also came from the Polyphone database, so it was recorded under similar conditions as the other two sets. In particular it was recorded over a telephone line while our recognizer should be able to recognize speech recorded by a PC microphone and it should be easily adaptable to specific tasks and applications.

Table 5: Evaluation Results

Test	Percentage of words recognized	Word accuracy percentage	Percentage of sentences correct
Develop. Set	95.27%	89.59%	38.43%
Eval. Set	93.55%	88.76%	32.56%
Other data	87.30%	84.59%	36.36%
Adapted	96.76%	95.41%	60.61%
4 person adapted	91.80%	93.44%	47.62%
Adapted, Polyphone input	32.16%	-21.44%	36.36%
Grammar	75.30%	72.59%	68.00%

To test how well the recognizer generalized to other data and other environments the data set we recorded ourselves was used. The data set was split in a training set containing about 500 utterances, that is, about 100 per person and a test set containing 30 utterances.

First the system was tested without any further training (second row in table 5). Although still reasonable the performance clearly decreased in comparison to the performance of the systems described in the last two paragraphs. These results were to be expected, since the data set was recorded using a video camera in a quiet laboratory, while the Polyphone data on which the system was trained was recorded over a normal telephone line. So the ambient noise, which is also modeled in the acoustic HMMs, will be quite different, which means that the acoustic HMMs will have slightly different distribution functions. As a result the acoustic vectors 'produced' by the HMMs are still similar to the acoustic vectors in the data set, but there is some distortion, causing classification mistakes. To make up for this effect, the system was adapted to the new situation by re-estimating twice, using the training part from the recorded data set.

As can be seen from table 5 the performance considerably increased. Actually, it is better than any of the results obtained in earlier tests; especially the word accuracy and the percentage of correct sentences showed a large improvement. The explanation for these results lies in the fact that the system not only adapted to the background noise in this data but it also adapted to the voices of these five persons. In fact the system has become speaker dependent.

The voices it adapted to it recognized very well, but now recognition of other speakers might give some trouble. To show these effects, two more experiments were performed. In the first experiment the system was adapted using only 4 different persons. Recognition was performed using data from the fifth person. The results show that although the performance is not as good as in the previous test, the system is still better than the undapted system. So the system adapted to the new environment but is still capable of generalizing and performing speaker independent recognition.

In the second experiment recognition was performed once again on the Polyphone test set using the system that was adapted to four persons. The recognition results were dramatic, showing that the system no longer recognized the data it was developed with; it completely adapted to the new environment. That the performance is this low is due to the fact that the utterance in the Polyphone database contains much more noise than the utterances used here, since they are recorded over a telephone line. Recognizing the PC recorded data with an unadapted Polyphone trained system worked because from the systems point of view these were just very high quality recordings. But form the point of view of the adapted system the Polyphone data set contains very noisy recordings, indeed many sounds were classified as mouth noise.

Our data set also contained sentences that adhered to a telebanking application grammar. This application allowed people to manage their bank accounts and conduct financial transactions by telephone. A language model was created that implemented this grammar. The last row of table one shows the results that were obtained from tests with this language model and the corresponding sentences from the data set. The adapted acoustic models were used.

One would expect the percentage of correct words to be higher as only a small vocabulary is used and the syntax of the sentences is constrained. A sentence by sentence inspection of the results showed what was actually going on here. In most cases the system did recognize the right sentence, but when it made a mistake, it often recognized a completely wrong sentence in which only a few words were correct, because the grammar forced the Viterbi algorithm to take a specific path. So about 25 percent of all sentences are responsible for most word errors.

CONCLUSIONS

In this paper we described the development of a large vocabulary speech recognizer for the Dutch language. The system was build using an incremental approach that started with a simple single Gaussian monophone system, which was refined in a number of steps. The final system uses backed-off triphones that solve the problem of unseen triphones without the need for specific linguistic

knowledge. The final system performs well, more than ninety percent of all words are recognized correctly. The errors that occur are usually small and typically involve wrong conjugations of a verb or hesitations by the speaker. A more powerful language model or a postprocessing module that checks the syntax of the sentences could reduce this kind of errors. The recognizer can cope with noises like smacking or loud breathing and the system is speaker independent. It has been tested with vocabularies of more than 5000 words, the performance decreased only slightly in this case. The system can be adapted to other environments and performance can be further improved, especially on sentence recognition, by making it person dependent.

REFERENCES

Boogaart, T.I., Bos L., Boves, L., *"Use of the Dutch Polyphone Corpus for Application Development"*, *Proceedings 2nd IEEE workshop on Interactive Voice Technology for Telecommunications Applications*, pp. 145-148, 26-27 September 1994, Kyoto, Japan

Damhuis M., Boogaart T., in 't Veld, C., Versteijlen, M.,W. Schelvis, W., Bos, L., Boves L., "Creation and Analysis of the Dutch Polyphone Corpus", *Proceedings International conference on Spoken Language Processing*, (ICSLP) '94, pp. 1803-1806, 18-22 September 1994,Yokohama, Japan

Grocholewski, S. "Acoustic Modeling for Polish*"*, *International Workshop Speech and Computers*, SPECOM, 2000

Jelinek, F., "Statistical Methods for Speech Recognition*"* (Language, Speech, and Communication) MIT Press, January 1999

Rabiner, L.R., Juang, B. H., "Fundamentals of Speech Recognition", Prentice Hall, Englewood Cliffs, N.J., 1993

Young, S., Kersaw, D., Odell, J., Ollason, D., Valtchev, V., Woodland, P. C., "The HTK Book" (for HTK Version 3.0*)*, Cambridge University Engineering Department

Wiggers, P. "Hidden Markov Models for Automatic Speech Recognition and their multimodal applications*"*, Master Thesis, Delft University of Technology, 2001, The Netherlands

Wojdel, J. "The Audio-Visual Corpus for Multimodal Speech Recognition in Dutch Language*"*, Internal report, Delft University of Technology, 2001, The Netherlands

Woodland P.C., Odell, J., Young S.J., "Tree-Based Tying for High Accuracy Acoustic Modelling", *Proceedings ARAPA Human Language Technology Workshop*, 1994

Woodland, P.C., Odell, Young, S.J., "Large Vocabulary Continuous Speech Recognition Using HTK*"*, *Proceedings International Conference on Acoustics Speech and Signal Processing*, 1994