Error detection in automatic transcriptions using Hidden Markov Models

Thomas Pellegrini[1], **Isabel Trancoso**[12]

[1]INESC-ID Lisboa, Portugal
[2]IST, Lisboa, Portugal
thomas@l2f.inesc-id.pt

Abstract

This article addresses error detection in broadcast news automatic transcription, as a post-processing stage. Based on the observation that many errors appear in bursts, we investigated the use of Hidden Markov Models (HMM) for their temporal modelling capabilities. Experiments were conducted on the 3 hour English broadcast news 1997 evaluation corpus from NIST. Common features in error detection were used, all decoder-based. Classification performance was compared with the discriminative maximum entropy model (Maxent) currently used in our in-house decoder to estimate confidence measures, and also with Gaussian Mixture Models (GMM). Our best HMM system detected 30.7% of the errors versus 28.2% with a GMM, and 14.8% for the Maxent model with a 0.5 decision threshold on the confidence measure estimates. To be compared to the standard ASR output, the best HMM system yielded to an improvement of the classification error rate from 17.6% to 16.6%.

Keywords: Error detection, automatic speech recognition

1. Introduction

Error detection is an important topic in Automatic Speech Recognition (ASR). Three types of errors can occur in the hypothesized word stream output: substitutions, insertions and deletions. Having a confidence measure indicating a potential substitution or insertion error for each hypothesized word is useful in several applications: to discard sentences with errors in real-time broadcast news subtitling systems, to try to correct errors by searching text material similar to what is being transcribed, to help select automatically material for unsupervised model training or speaker model adaptation, to validate results of keyword spotting, or else to detect out-of-vocabulary words.

Confidence measures can be used to classify hypothesized words into two classes, "correct" and "error". Many statistical tools have been proposed in the literature: generalized linear models (Gillick et al., 1997, Allauzen, 2007), artificial neural networks (Weintraub et al., 1997) and more recently conditional random fields (Xue et al., 2006). Confidence estimation is still challenging, since one of the difficulties remain in the decoding process itself: to allow computation efficiency, the search space is pruned. Hence, word posteriors that are the main feature for confidence estimates are over-estimated (Hillard et al., 2006).

This problem will not be addressed in this article, rather we will focus on a common observation, that errors appear very often in bursts. For example, an out-of-vocabulary word is known to generate between 1.5 and 2 errors (Schwartz et al., 1994). Error bursts are well illustrated in the following alignment example, between our ASR decoder output and the corresponding reference. The named entity "John Makero" was not part of the recognition vocabulary and appears to be responsible of three consecutive errors, indicated by surrounding stars:

```
ref:   DR. *JOHN** *MAKERO* *IS*** A PROFESSOR
hyp:   DR. *ZHANG* *MARKET* *ROSE* A PROFESSOR
```

The presence of multi-word error sequences in the output word stream justifies the use of statistical tools that model temporal sequences in some way, such as Hidden Markov Models (HMM), or linear-chain conditional random fields. In this study, we propose a two-state HMM, with one "error" state, and one "correct" state, respectively trained on only errors and correct words from the decoder output.

In the following, features for error modelling will be presented and the choice of Hidden Markov Models will be discussed. Section 4 describes the American English HUB-4 NIST evaluation corpus used to train and test the models. Then error detection results are provided, based on the automatic transcription of the corpus performed by our in-house decoder. HMM classification results will be compared to results achieved by a Gaussian mixture model, and a maximum entropy model, trained on the same data.

2. Features for error detection

The output of the ASR system is a stream of words. For each hypothesized word, various decoder-based features are available. In this study, only words from the best hypothesis are considered.

A set of 15 features common in error detection was used:

. Length of words in number of decoding frames (20 ms duration) and in number of phones (2)
. Final, acoustic and posterior scores (3)
. Average phone acoustic and posterior scores (2)
. Log of the total and average active states, arcs and tokens (6)
. Minimum and average phone log-likelihood ratios (2)

Features related to the active states, arcs and tokens for each hypothesized word should be intuitively high to reflect a large degree of uncertainty of the recognizer (Gillick et al., 1997).

3. Models for error detection

Many distinct types of statistical classifiers can be used. Currently, our in-house ASR system estimates confidence measures with a maximum entropy model. In this study, we compared this discriminant model with generatives models, Gaussian Mixture Models and Hidden Markov Models.

3.1. Maximum Entropy models

Maximum Entropy (Maxent) models are very popular models, and are used in many applications, in particular in natural language processing tasks, such as part-of-speech tagging (Ratnaparkhi, 1996). The Maxent principle states that the correct probability distribution for a given class is the one that maximizes entropy, given constraints on the distribution (Jaynes, 1957). One advantage of Maxent models is that the training algorithm will determine how to combine the different features by estimating the best weights, so that the main user effort will consist of identifying which features are best to be used. In our case, the Maxent was used as the following: when the probability or confidence measure given by the model is lower than 0.5, then the hypothesized word is labeled as an error. In pratice, larger decision thresholds are used: about 0.8 and more to select automatically transcribed data to do unsupervised acoustic model training for example. To train the Maxent model, the Megam toolbox[1] was used.

3.2. Hidden Markov Models

In Hidden Markov Models (HMM), a sequence of hidden (not observable) states generates a sequence of observations, where each state has its own probability distribution, generally a mixture of Gaussians (Rabiner et al., 1986). HMMs are very adapted to compute a probability of a sequence of temporal observations. For that reason HMMs appear very attractive to detect error sequences. The HTK toolbox[2] has been used to train and test HMMs.

The HMM scheme with the transition probabilities is shown in figure 1. A 2-state HMM is used, with one "error" state and one "correct" state. To train this HMM, one would need to align the training data at state-level, which does not make sense since by definition, states are hidden. Hence, each state was trained separately as single HMM and then merged together into the final model. This approach allows to use different numbers of Gaussian mixtures for both states according to the available amount of training data for each class.

The transition matrix is designed manually. Since errors often occur in bursts, the self-loop probability to stay in the single *error* state (value 0.55), has been chosen larger than the transition probability between the two states (value: 0.35). Also, since there are much less errors than correct words, we applied the same transition values for the *correct* state. Intuitively, it is more likely to have a correct word if the preceding one is correct.

[1]available at http://www.cs.utah.edu/~hal/megam

[2]available at http://htk.eng.cam.ac.uk

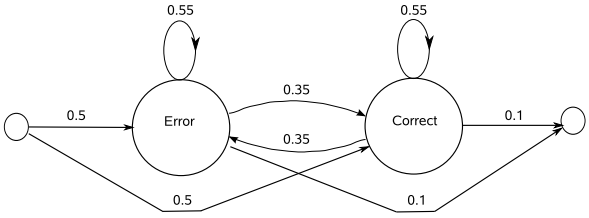

Figure 1.: HMM used for error detection. One state models the *error* probability distribution, and the other state models the *correct* distribution. Self-loop probabilities, ie probabilities to stay in a same state, have been chosen slightly larger than the transition probabilities between the two states to model the observed bursts of errors. The two smallest circles in the figure are the entry and exit non-emitting states.

3.3. Gaussian Mixture Models

Gaussian Mixture Models (GMM) are linear combinations of Gaussian probability densities, whose weights, means and variances are optimized on a training corpus (Bishop et al., 2006). GMMs can be seen as single state HMM. Hence, GMM have no temporal modelling capabilities. It is interesting to compare their performance to HMMs, to evaluate the need to model sequences. In this study, two GMMs were trained, one for the *error* class and one for the *correct* class. Classification is made based on a simple comparison between the log-likelihoods estimated with the two models.

4. Corpus

The corpus used in this study is the 1997 evaluation corpus of the NIST HUB-4 American English transcription campaign. It is part of the LDC catalog, under the reference LDC2002S11[3]. This broadcast news test set totalizes about 3 hours of manually transcribed speech.

The three hours were transcribed automatically with our in-house speech recognition decoder, that will be briefly described in section 5.2. This material was divided into a training and test corpus, the test corpus corresponding to about 10% of all the automatic transcriptions in number of words. Table 1 gives the number of transcribed words for both subsets. Since transcription errors are our classification target, errors were considered as "positive" examples.

Train		*Test*	
Total	26,593	*Total*	3,437
Positives	*Negatives*	*Positives*	*Negatives*
4,669	21,924	600	2,837

Table 1.: Number of positive (errors) and negative (correct words) examples in both train and test sets.

5. Experiments

5.1. Evaluation

Errors are detected only with hypothesized words, thus only substitutions and insertions are addressed, and not deletions. Hence, the Word Error Rate (WER) is given by:

[3]available at www.ldc.upenn.edu

$$\text{WER} = \frac{\text{\# (Substitutions+Insertions)}}{\text{\# (hypothesized words)}}$$

Error detection will be evaluated on a global Classification Error Rate (CER), defined as:

$$\text{CER} = \frac{\text{\# (Number of incorrect classifications)}}{\text{\# (hypothesized words)}}$$

Nevertheless, CER depends on the relative sizes of the number of errors and correct words. Since there are many more correct words than errors, CER is not very satisfying to measure error detection performance. Hence, classifiers will also be characterized by statistics over true and false positives, in particular by drawing Receiver Operating Characteristics (ROC).

5.2. Baseline

Experiments were conducted on our in-house ASR system, named AUDIMUS, a hybrid Artificial Neural Networks / Hidden Markov Models system (Meinedo et al., 2003). A set of 455 context dependent diphone-like acoustic models, plus two non-speech models (one for silence and one for breath) is used to transcribe American English. More details about the context dependency modelling can be found in (Abad et al., 2008). Acoustic models were trained on 140 hours of manually transcribed HUB-4 speech. The language model is a 4-gram model, with Kneser-Ney modified smoothing, trained on 150 million words from HUB-4 transcripts, and about 1 billion words of newspaper and newswire texts. The 64k word vocabulary consists of all the words contained in the HUB-4 training set plus the most frequent words in the broadcast news texts and Newspapers texts. Multiple-pronunciations are allowed and totalize 70k entries.

The word error rate (WER) for the entire corpus is 17.6%. This value is higher than the official WER when a normalization on the output is used before scoring (verbal form expansion in particular). This WER can be seen as our baseline, being the classification error rate of an ultra-liberal classifier, that would predicts as correct all the output words.

5.3. Error detection results

First, a Maxent model, a GMM and a HMM have been trained on the same training data set. Second, based on the confidence measure estimates provided by the Maxent model, a training subset was extracted by selecting training examples with a Maxent confidence measure lower than 0.6 for errors, and larger than 0.7 for correct words. This resulted in about four times fewer error training examples (1.3k examples), and only 10% relative fewer correct training examples (19.7k examples). Test set was unchanged of course. The henceforth called 'GMM-c' and 'HMM-c' ('c' for constrained) were trained on this data. All GMMs and HMMs have 512 and 32 Gaussian mixtures for respectively the *correct* state and the *error* state. In mean, this gives respectively about 45 and 150 examples to train the two models. Larger numbers of mixtures were tried for the *error* model, but lead to worse results.

Figure 2 shows the ROC graph of the Maxent model and the two HMMs, achieved on the test data. The thicker plain line gives the decision threshold with which the Maxent curve was drawn, as a function of the false alarm rate. In this type of graphs, points closer to the left-top corner correspond to the best classifiers. On the ROC curve, our baseline, the ultra-liberal classifier is located at the graph origin, since the numbers of detected errors, i.e. true positives and false alarms, are equal to zero.

Table 2 shows the corresponding classification results, the *error* class being considered as the positive class. The table gives the Classification Error Rate (CER), along with positive and negative detection statistics: true and false positives and negatives.

The Maxent model outperforms the baseline, with a 15.9% CER. Nevertheless, only 89 errors out of 600 have been detected. Most of the probabilities given by the Maxent model are larger than the standard 0.5 threshold, even for erroneous words. This threshold can be chosen larger but the CER will increase, due to larger number of false alarms. According to the ROC curve, that focus only on positive performance rates, the best working point would be the (false alarm rate=22.1%, detection rate=63.0%) point closest to the (0,1) ideal point, corresponding to a 0.81 decision threshold. At this working point, Maxent detects about 380 true positives, but the number of false alarms is about 630. The corresponding CER is 24.7%, much higher than the baseline.

Globally, the HMM trained on all the training data performed worse than the Maxent model with a 0.5 threshold in terms of CER, with a 17.3% CER. The 'HMM' point on the ROC curve is indeed below the Maxent ROC curve. Nevertheless, the number of detected errors is about 35% larger than for Maxent, with 120 true positives.

The 'HMM-c' classifier yielded better results than 'HMM'. The false alarm rate increased in comparison to 'HMM' much less than the true positive rate increased. 184 errors were correctly detected, resulting in a 16.6% global CER. To be compared with Maxent, 'HMM-c' shows similar performances when using a 0.65 decision threshold with the Maxent model. 96.5% of the *error* predictions made by these two models are the same.

Finally, both GMMs perform worse than their corresponding HMMs. Less true positives were detected, and larger CER were observed with GMMs. The temporal modelling capability of HMMs may explain its superiority over GMMs.

5.4. Result analysis

All classifiers present high false alarm rates. In particular, when increasing the decision threshold used with the Maxent model, the number of wrong *error* label detections (false alarms) rapidly becomes very high. The most frequent false alarms appear to be very short words. The ten most frequent false alarms are: *THE, IN, I, TO, SOME, OF, A, THIS, AS, AND*. The mean word length in characters of the false alarms is smaller than the mean length for the true positives: 4.9 versus 6.1. This may be due to the fact that most insertion errors of the decoder are small words. Then, the error classifier tends to label short words as errors too easily. When using

	CER	tp	fp	tn	fn
baseline	17.6	0	0	2,837	600
Maxent	15.9	89	37	2,800	511
GMM	18.1	131	153	2,684	469
GMM-c	17.0	169	153	2,684	431
HMM	17.3	120	116	2,721	480
HMM-c	16.6	184	156	2,681	416

Table 2.: Results in terms of classification error rate (CER), true and false positives (tp, fp) and negatives (tn , fn) for the baseline ("ultra-liberal classifier"), a Maxentropy model (with a 0.5 decision threshold), two Gaussian Mixture Models (GMM), and two Hidden Markov Models (HMM), trained on different data: 'GMM' and 'HMM' were trained with the same material as the Maxent model, 'GMM-c' and 'HMM-c' with a subset comprised of the samples with a confidence measure lower than 0.6 for the errors, and larger than 0.7 for the correct samples.

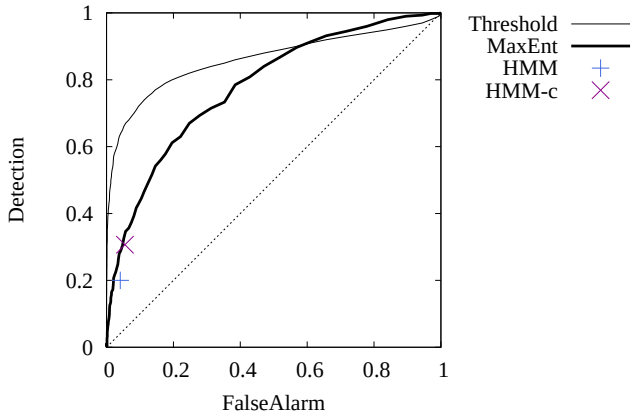

Figure 2.: ROC graph achieved on the test set with the Maxent model and two HMM. The threshold curve corresponds to the confidence measure threshold used to draw the Maxent curve.

confidence measures, a higher decision threshold could be used for frequent short words.

Globally, 'HMM-c' and 'Maxent' with a 0.65 threshold predict the same errors: 96.5% of the *error* predictions are the same. It is interesting to characterize the few prediction differences. Figure 3 shows the number of word sequences correctly labeled by 'Maxent' and 'HMM-c', as a function of their length in number of words. It appears that 'Maxent' predicts slightly more single word errors and less multi-word error segments than 'HMM-c'. For example, 'HMM-c' correctly labeled 18 two-word segments as error versus only 12 for 'Maxent'. This difference is very small, but shows a slightly better capability of HMM to detect multi-word sequences of errors.

5.5. Impact of the transition matrix probabilities

One intuition that lead to test HMMs to detect ASR output errors was that very often, errors appear in bursts. The self-loop probability to stay in the *error* state was therefore chosen larger than the transition to the other state. The transition probability matrix used so far in this study was:

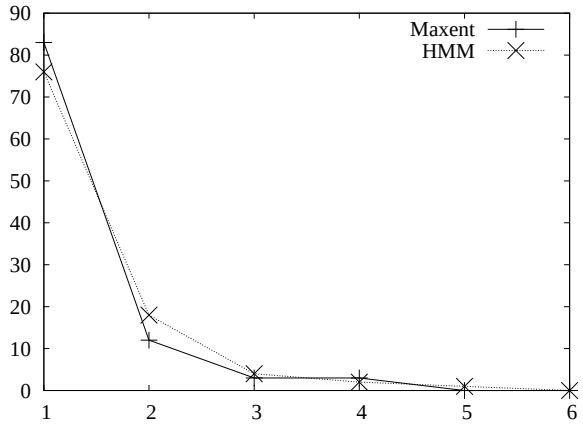

Figure 3.: Number of error segments correctly labeled by 'Maxent' and 'HMM-c', as a function of their length in number of words.

$$\begin{bmatrix} 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & \mathbf{0.55} & 0.35 & 0.1 \\ 0.0 & 0.35 & \mathbf{0.55} & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

Note that only the 2x2 central part of the matrix is of interest since the other values concern the entry and exit non-emitting states.

Results for two additional HMM, named 'HMM-c2' and 'HMM-c3', are reported here. Respective transition matrices are:

$$\begin{bmatrix} 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & \mathbf{0.45} & \mathbf{0.45} & 0.1 \\ 0.0 & \mathbf{0.45} & \mathbf{0.45} & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

and

$$\begin{bmatrix} 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.3 & \mathbf{0.6} & 0.1 \\ 0.0 & \mathbf{0.6} & 0.3 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

For 'HMM-c2', the probabilities to stay in a single state or to jump to the second state are the same (0.45). For 'HMM-c3', the probability transition between the two states is larger than the self-loop probability (0.6 opposed to 0.3).

Table 3 gives the performance results for the three models 'HMM-c', 'HMM-c2', 'HMM-c3', all trained on the same restricted data. Classification error rates for 'HMM-c2' and 'HMM-c3' are slightly larger than 'HMM-c' for which the results have been reported so far in this paper. In particular, the number of false alarms is larger for both systems. Other values for the transition matrix probabilities not reported here were tested, and transition values smaller than self-loop values always yielded better results. These results seem to validate the assumption that it is more likely to stay in a single state, *error* or *correct*.

6. Summary and future work

In this paper, the problem of error detection in automatic transcriptions has been addressed with the use of Hidden

	CER	tp	fp	tn	fn
baseline	17.6	0	0	2,837	600
HMM-c	16.6	184	156	2,681	416
HMM-c2	16.7	185	158	2,679	415
HMM-c3	16.8	193	170	2,667	407

Table 3.: Results in terms of classification error rate (CER), true and false positives (tp, fp) and negatives (tn , fn) for 'HMM-c2' and 'HMM-c3' that differ from 'HMM-c' only on the transition and self-loop probabilities assigned to the two *error* and *correct* states. Performance of the baseline and the 'HMM-c' classifier are reported to allow comparison.

Markov Models, with the idea that recognition errors often appear in "bursts", i.e. in sequences of several wrong hypothesized words. The HMM ability to model temporal sequences has been tested by comparing this approach to a Gaussian Mixture Model (GMM), and to a maximum entropy model, that is currently used in our in-house ASR system to estimate confidence measures.

Experiments were carried out on a three hour NIST evaluation test set of American English broadcast news speech. A Maxent model with a 0.5 decision threshold was able to detect correctly 89 errors over 600, with a global CER of 15.9%. A first HMM model, trained on the same data, showed worse performance, with a 17.3% CER.

In a second experiment, the training material was reduced by constraining the training examples based on the Maxent confidence measure estimates: only errors with a probability smaller than 0.6 and correct words with a probability larger than 0.7 were selected from the training set. This data selection yielded a significant improvement of the CER from 17.3% to 16.6%, with 184 true errors detected. This HMM performance has been found equivalent to the Maxent performance when using a 0.65 threshold on the confidence measures. One advantage to use HMMs is that there is no need to choose an arbitrary threshold as it is the case with the Maxent approach.

It has been shown also that the most frequent false alarms involve very short words. Indeed, a lot of recognition errors made by the decoder involve short words that are more easily inserted or substituted than longer words. One idea that remains to be tested would be to discard error training examples for very short words that have a high confidence measure. More generally, the 0.6 and 0.7 thresholds used to constrain the training data could be optimized on a development set.

Result analysis showed that the Maxent model detected almost only single error words in the decoder word output stream. The HMM system was able to detect more multi-word error sequences, justifying the use of a model with temporal sequence modelling capabilities. This last assumption has been also confirmed by the HMM superiority over GMMs. Future work will consist in comparing HMMs to their somehow equivalent discriminant model, linear-chain conditional random fields, recently used in detecting errors, and more generally in segmenting and labelling sequence data.

Finally, the ability to mark words recognized with low confidence in an automatically recognized broadcast news transcript is also very relevant for our computer aided language learning system (Marujo et al., 2009). Learning from recent documents such as broadcast news videos with automatically produced captions is one of the features that may make the use of the system more motivating for students. The captions have a different color for the low confidence words.

7. Acknowledgement

This work was partially funded by the REAP.PT project (CMU-PT/HuMach/0053/2008) and the Vidivideo European project.

References

Gillick, L. Ito, Y. and Young, J. (1997) *A probabilistic approach to confidence estimation and evaluation*, in proceedings of ICASSP, Munich, pp. 879-882.

Allauzen, A. (2007) *Error detection in confusion*, in proceedings of INTERSPEECH, Antwerp, pp. 1749-1752.

Weintraub, M., Beaufays, F., Rivlin, Z., Konig, Y., and Stolcke, A. (1997) *Neural - Network Based Measures of Confidence for Word Recognition*, in proceedings of ICASSP, Los Alamitos, pp. 887-890.

Xue, J. and Zhao, Y. (2006), *Random forests-based confidence annotation using novel features from confusion network*, in proceedings of ICASSP, Toulouse, pp. 1149-1152.

Hillard, D., and Ostendorf, M. (2006) *Compensating for Word Posterior Estimation Bias in Confusion Networks*, in proceedings of ICASSP, Toulouse, pp. 1153-1156.

Schwartz, R., Nguyen, L., Kubala, F., Chou, G., Zavaliagkos, G., and Makhoul, J. (1994), *On Using Written Language Training Data for Spoken Language Modeling*, in proceedings of ACL, New Jersey, pp. 94-97.

Ratnaparkhi, A. (1996) *A Maximum Entropy Model for Part-Of-Speech Tagging*, in proceedings of EMLNP, Philadelphia, pp. 133-142.

Jaynes, E.T. (1957), *Information theory and statistical mechanics*, Physical review, Vol. 106:4, pp. 620-630.

Rabiner, L.R. and Juang, B.H. (1986), *An Introduction to Hidden Markov Models*, in IEEE Acoustics Speech and Signal Processing Magazine, ASSP-3(1), pp. 4-16.

Bishop, C. (2006), *Pattern recognition and machine learning*, Springer.

Meinedo, H. and Caseiro, D. and Neto, J. and Trancoso, I. (2003), *AUDIMUS.media: a broadcast news speech recognition system for the european portuguese language*, in proceedings of PROPOR, Faro, pp. 9-17.

Abad, A. and Neto, J. (2008), *Incorporating Acoustical Modelling of Phone Transitions in a Hybrid ANN/HMM Speech Recognizer*, in proceedings of INTERSPEECH, Brisbane, pp. 2394-2397.

Marujo, L., Lopes, J., Mamede, N., Trancoso, I., Pino, J., Eskenazi, M., Baptista, J., and Viana, C. (2009), *Porting REAP to European Portuguese*, in SLATE 2009 - Speech and Language Technology in Education, Brighton.