Noname manuscript No.
(will be inserted by the editor)

Error detection in broadcast news ASR using Markov chains

Thomas Pellegrini · Isabel Trancoso

Received: date / Accepted: date

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 Markov Chains (MC) for their temporal modelling capabilities. Experiments were conducted on a large Amercian English broadcast news corpus from NIST. Common features in error detection were used, all decoder-based. MC classification performance was compared with a discriminative maximum entropy model (Maxent), currently used in our in-house decoder to estimate confidence measures, and also with Gaussian Mixture Models (GMM). The MC classifier obtained the best results, by detecting 16.2% of the errors, with the lowest classification error rate of 16.7%. To be compared with the GMM classifier, MC allowed to lower the number of false detections, by 23.5% relative. The Maxent system achieved the same CER, but detected only 7.2% of the errors.

Keywords Error detection · automatic speech recognition · Markov chains

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.

T. Pellegrini
INESC-ID Lisboa
R. Alves Redol, 9
1000-029 LISBON, Portugal
Tel.: +351 213-100-268
Fax: +351 213-145-843
E-mail: thomas.pellegrini@l2f.inesc-id.pt

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 our recognition vocabulary and appeared 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 Markov Chains (MC), or linear-chain conditional random fields. In this study, we propose a two-state MC, 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 listed, and the various statistical models will be briefly presented. Section 4 describes the American English HUB-4 NIST 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. Classification results of the various classifiers will be compared, and complementary experiments with MC will be presented.

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 generative models, Gaussian Mixture Models and Markov Chain 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 Markov Chains

In Markov Chains (MC), a sequence of observable states generates a sequence of observations. Each state has its own probability distribution, generally a mixture of Gaussians [Rabiner et al.(1986)]. MCs are very adapted to compute a probability of a sequence of temporal observations. For that reason MCs appear very attractive to detect error sequences.

The MC scheme with the transition probabilities is shown in figure 1. A 2-state MC is used, with one “error” state and one “correct” state. To train this MC, 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 MC 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 was 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. The HTK toolkit[2] was used to train and test MCs.

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

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

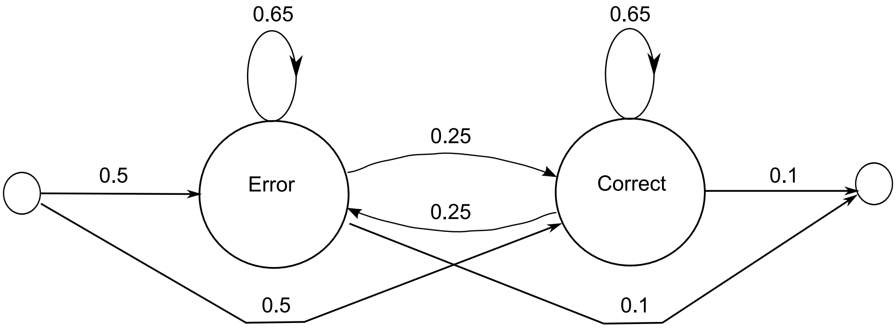

Fig. 1 Markov chain used for error detection. One state models the *error* probability distribution, and the other state models the *correct* distribution. Self-loop probabilities, ie probabilities to stay in a same state, have been chosen larger than the transition probabilities between the two states to model the error burst phenomenom. The two smallest circles in the figure are entry and exit non-emitting states.

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

Train		*Test*	
Total	674,452	*Total*	30,014
Positives	*Negatives*	*Positives*	*Negatives*
57,326	617,126	5,192	24,822

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 MCs. Hence, GMM have no temporal modelling capabilities. It is interesting to compare their performance to MCs, 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 were the training and evaluation corpora of the NIST HUB-4 American English transcription 1997 campaign[3]. These broadcast news sets totalizes respectively about 67 hours and 3 hours of manually transcribed speech.

The 70 hours were transcribed automatically with our in-house speech recognition decoder, that will be briefly described in section 5.2. All the training recognized data was used to train our statistical classifiers. Table 1 gives the number of transcribed words for both data sets. Since transcription errors are our classification target, errors were considered as the "positive" class.

[3] These corpus, with references LDC1998S71 and LDC2002S11, are available at `www.ldc.upenn.edu`

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:

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

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 hopefully 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 Automatic transcription

Experiments were conducted with 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 test corpus was 24.0%. This value is higher than the standard WER, when a normalization on the output is used before scoring (verbal form expansion in particular). The WER can be seen as the classification error rate of an ultra-liberal classifier, that would predicts as correct all the output words.

5.3 Error detection results

Maxent models, GMMs and MCs were trained on the same data set. All GMMs and MCs have 512 and 32 Gaussian mixtures for respectively the *correct* state and the *error* state. In mean, this gives respectively about 1200 and 1800 examples per mixture to train the two models. Larger numbers of mixtures were tried for the *error* model, but lead to worse results.

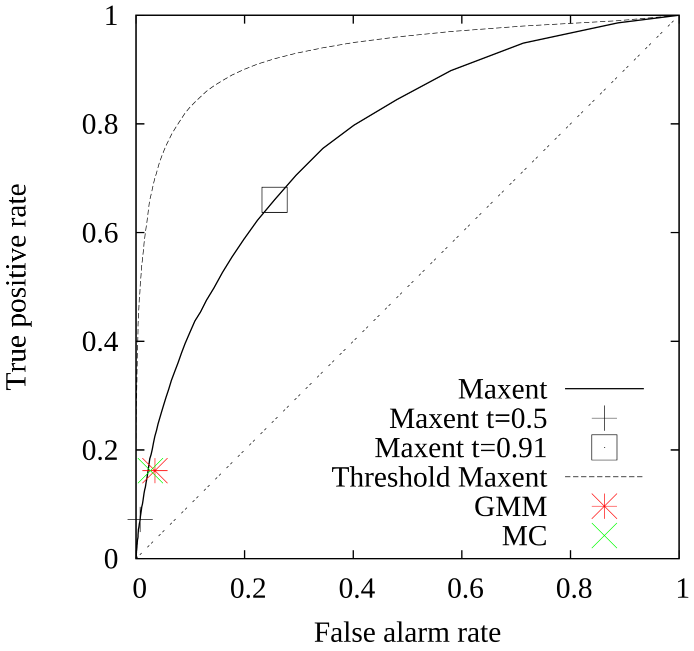

Fig. 2 *ROC graph. The threshold curve corresponds to the confidence measure threshold used to draw the Maxent curve.*

Table 2 *Results in terms of classification error rate (CER), true and false positives (tp, fp) and negatives (tn , fn).*

	CER	tp	fp	tn	fn
Maxent t=0.5	16.7	375	189	24,633	4,817
Maxent t=0.91	25.0	3,233	5,554	19,268	1,959
GMM	17.4	840	863	23,959	4,352
MC	16.7	840	660	24,162	4,352

Table 2 shows the 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 negatives and positives.

To further illustrate these results, figure 2 shows the ROC graph for the different classifiers. In this type of graph, points closer to the left-top corner correspond to the best classifiers. The datched line gives the decision threshold with which the Maxent curve was drawn, as a function of the false alarm rate. For example, when a 0.8 threshold is used, all confidence measures estimated by Maxent that are smaller than 0.8 will give an *error* label for the words to be classified. The plain line corresponds to the performance of the Maxent classifier, when varying the decision threshold. Two points on the Maxent curve were added to show the Maxent performance, at two operating points: one corresponding to a standard 0.5 threshold value, and one to the best threshold value of 0.91 as predicted by the curve, to be as close to the top-left corner as possible. The Maxent results can be seen as our baseline, since it is actually used in our ASR system. Performance of the GMM and the MC classifiers are indicated by only two single points, since they were used as binary decision tools, by simply comparing the probabilities for a word to be correct or wrong.

Maxent with a 0.5 threshold value and MC gave the best CERs, with a 16.7% value. Nevertheless, only 375 errors out of 5.2k were detected by Maxent, whereas MC detected 840 errors, corresponding to 16% of the errors. Most of the probabilities given by the Maxent model were larger than the standard 0.5 threshold, even for mis-recognized words. Thus, this threshold can be chosen larger but the CER will increase, due to a 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=25.5%, detection rate=66.0%) point closest to the (0,1) ideal point, corresponding to a 0.91 decision threshold. At this working point, Maxent detected about 3.2k true positives, but the number of false alarms was very high, with a 5.6k value. The corresponding CER was 25.0%, which is larger than the WER.

80% of the errors detected by GMM were also detected by MC, but GMM showed a much larger number of false alarms, with an increase of about 30% relative. Both GMM and MC ROC points are much higher in the graph than the Maxent t=0.5 point, showing better results. Nevertheless, these points are still far from the ideal (0,1) point. The choice of using MC or Maxent at t=0.91 will depend on the application. If priority is given to detection, then large number of false detections may not be critical, and Maxent with a large threshold could be used. In the inverse case, MC might be better.

5.4 Result analysis

All classifiers presented 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 increased. The most frequent false alarms (FA) appeared to be very short words. The ten most frequent FA were: *THE, IN, I, TO, SOME, OF, A, THIS, AS, AND*. The mean word length in characters of the false alarms was 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 were small words. Then, the error classifier was inclined to label short words as errors too easily. When using confidence measures, a higher decision threshold could be used for frequent short words.

It is interesting to compare GMM and MC, since GMM can be seen as a single-state MC, with no temporal modelling capabilities. The better performance of MC was due to a smaller number of FAs, compared to GMM. These false detections correspond mainly to single error labels in a sequence of correct words, instead of error detections in a sequence of consecutive errors in the ASR output. This seems to confirm the usefullness of the temporal modelling capabilities of MC, which give more “inertia” to the model. When the preceding word has been classified as correct, the current word has a higher probability to be also labeled as correct, and idem for the error class. Figure 3 shows the number of word sequences correctly labeled by Maxent, GMM and MC, as a function of their length in number of words. It appears that MC predicted more multi-word error sequences than the other two models. GMM predicted much more single-word errors.

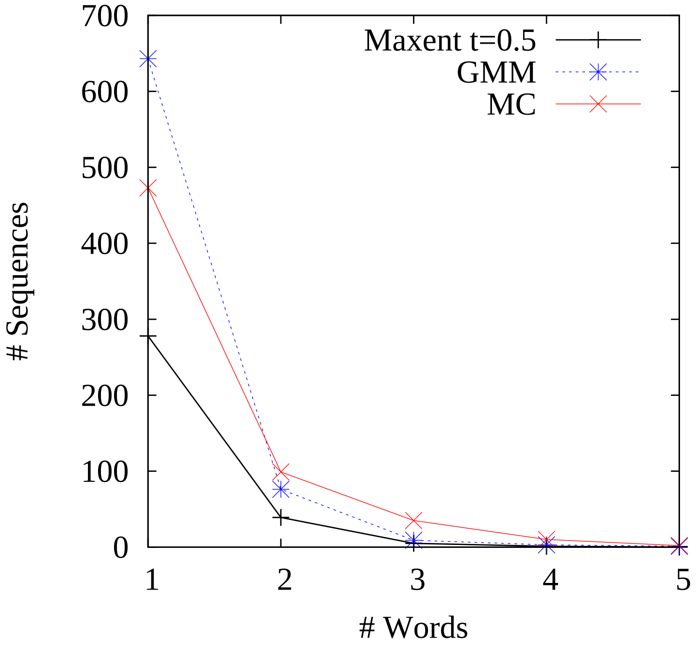

Fig. 3 Number of error segments correctly labeled by Maxent, GMM and MC, as a function of their length in number of words.

5.5 Impact of the transition probability matrix

One intuition that lead to test MCs 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:

$$\begin{bmatrix} 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & \mathbf{0.65} & 0.25 & 0.1 \\ 0.0 & 0.25 & \mathbf{0.65} & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

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 MC, named hereafter MC-b and MC-c, are reported here. Respective transition matrices were the following:

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

and

Table 3 *Classification error rate (CER), true and false positives (tp, fp) and negatives (tn , fn) for MC-b and MC-c that differ from MC only on the transition and self-loop probabilities.*

	CER	tp	fp	tn	fn
MC	16.7	840	660	24,162	4,352
MC-b	17.3	1,114	1,124	23,698	4,078
MC-c	18.7	1,443	1,877	22,945	3,749

$$\begin{bmatrix} 0.0 & 0.5 & 0.5 & 0.0 \\ 0.0 & 0.25 & \mathbf{0.65} & 0.1 \\ 0.0 & \mathbf{0.65} & 0.25 & 0.1 \\ 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

For MC-b, the probabilities to stay in a single state or to jump to the second state are the same (0.45). For MC-c, the probability transition between the two states is larger than the self-loop probability (0.65 opposed to 0.25).

Table 3 gives the performance results for the three models MC, MC-b, MC-c, all trained on the same data. Classification error rates for MC-b and MC-c were larger than MC's CER. The number of correct detections increased, but the number of false alarms increased much more. Self-loop probabilities larger than transition probs gave the best performance. This result seems to validate the assumption that it is more likely to stay in a single state, *error* or *correct*, i.e. errors mostly occur in bursts, rather than isolated.

6 Summary and future work

In this article, the problem of detecting errors in automatic transcriptions has been addressed with the use of various statistical tools, with the idea that recognition errors often appear in "bursts", i.e. in sequences of several wrong hypothesized words. The Markov chains 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 large American English broadcast news speech NIST corpus. A Maxent model with a 0.5 decision threshold was able to detect only 7% of the errors correctly. With a 0.91 threshold, this percentage raised 62% but the number of false errors increased much more, to almost double the number of correct detections. The Markov chain outperformed Maxent and GMM, with a 16.7% CER and 860 errors correctly detected. The temporal modelling capabilities of this model seemed to bring a useful "inertia" that lowered the number of false alarms. The choice of using MC or Maxent with a large decision threshold will depend on the application. If priority is given to detection, then large number of false detections may not be critical, and Maxent could be used. In the inverse case, MC might be better.

Result analysis showed that the Maxent and GMM models detected mainly single-error words in the decoder word output stream. The MC 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 MC superiority over GMM. Future work will consist in comparing MCs to their somehow equivalent discriminant model, linear-chain conditional random fields, recently used in many natural language processing tasks.

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 Acknowledgements

This work was partially supported by FCT (INESC-ID multi-annual funding) through the PIDDAC Program funds, by the FCT REAP.PT project (CMU-PT/HuMach/-0053/2008) and the Vidivideo European project.

References

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

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

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

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

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

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

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

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

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

[Bishop et al.(2006)] C. Bishop. (2006), *Pattern recognition and machine learning*, Springer.

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

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

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