Evaluation of Feature Extraction and Acoustic Modeling Sub-Systems by interfacing ASR Systems

Joan Marí Hilario, José-Manuel Ferrer Ruiz and Fritz Class

Speech Understanding Systems Department
DaimlerChrysler Research & Technology Center Ulm, Germany
joan.mari_hilario@daimlerchrysler.com

Abstract

The main task of DaimlerChrysler in RESPITE is to develop a demonstrator ASR system for the most successful robust techniques developed during the project. We intend to build the demonstrator by interfacing our ASR system with the systems that already employ them. Interfaces have been designed at two levels: feature and state-likelihood. Results of the validation of the interfaces are presented for the AURORA-2000 digit task.

1. Introduction

The final object of our work in the RESPITE is to develop a demonstrator system for the most successful robust ASR techniques developed during the project. We intend to do this by interfacing on-line the systems where the new robust ASR techniques are implemented, with our ASR system. Two interfaces have been defined: one at the 'feature' and the other at the 'state-likelihood' level. Since both interfaces must be tested and just the most successful techniques must be incorporated into this demonstrator, our job has an evaluative component too. As a result, the first part of this contribution is devoted to ASR system evaluation concepts, and how they can be useful to design a meaningful evaluation exercise. In the next point, our evaluation framework is presented using the evaluation concepts explained. Finally the results of our first experiments are shown and discussed.

1.1. Technology Assessment

The term[1-2] refers to a kind of assessment of mainly 'user-transparent' tasks - whose inputs/outputs are of no direct use for the user - or components of a Language Processing system that tends to ignore user-centred factors. Therefore this kind of assessment defines abstract criterion and measures, and apply them to LP system components without regard to user issues. Typically, a user-transparent task is part of a wider user-visible task, although this distinction is sometimes artificial, as what is user-visible for a certain given task, could be a user-transparent sub-task in another task, e.g. the output of an ASR system is certainly user-relevant in a Speech-to-Text system, but of no importance for the user of a Machine Translation system. Furthermore, it is even the case for technology assessment of ASR tasks that ASR systems are assessed in isolation, without any regard to the fact that they are a part of a LP system in any practical application.

1.2. Comparative Technology Assessment

But to be able to compare the results of different system or component assessments, certain conditions should be met. This leads us to 'comparative technology assessment', that will consist of fixing and letting vary across all system or component assessments certain attributes of the task, system/component implementation and environment - depending on the evaluation criteria and the degree of desired comparability - and 'compare' the assessment results of the systems or components. By varying or fixing any of those attributes one can have different kinds of evaluation exercises. For example, NIST/ARPA ASR[3] evaluations are characterised by having the environment and task fixed, while letting vary task attributes and implementation. However, it is important to keep in mind, that the assessment results of the systems should be kept comparable, which certainly limits the amount of allowed variability.

Another key point regarding comparability, meaning and usefulness of the results is to use common 'evaluation data'. This is normally composed of three parts, namely training , test – input - and reference – output -, and for a useful comparative assessment must certainly be 'realistic' and 'representative' for the task being evaluated. The former means that it must be the kind of data that the systems would actually process in real use, while the latter means that it should contain instances of the full range of inputs that the systems would receive.

To get comparable results it is certainly very important too to use common 'evaluation criteria', 'evaluation measures', and to apply them with a coherent 'evaluation methodology'. As pointed out at the beginning of the paragraph evaluation criteria determine the attributes to be fixed or varied. Consequently they are the first thing to be specified in any evaluation exercise. The following point is devoted to discuss those concepts.

1.3. Criteria and measures for ASR system evaluation

Since ASR systems are usually assessed in a isolated way, the prevalent abstract criterion and associated measure used to evaluate ASR systems is recognition performance and Word Error Rate (WER), although other criteria and measures like processing speed and real time factor are being introduced too, specially because ASR systems must be able to work in real-time if they are to be used in any realistic LP application. It's possible to define other measures of recognition performance too, like for instance the so called Relative Information Loss[4], but WER is by far the most used. By doing so, it's implicitly assumed that WER shows some correlation to our *de facto* measures, i.e. the ones applied to the overall Language Processing system of the intended application. Some authors[5] have indeed demonstrated that WER is a measure that strongly correlates with other Language Processing performance measures, as their

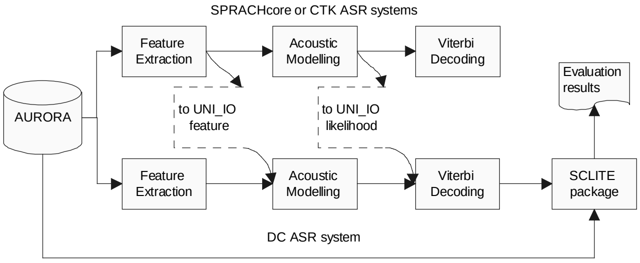

Figure 1Diagram of the evaluation using interfaces

experiments with it and Concept Accuracy indicate. An example of an evaluation that regards ASR systems as a part of a LP system is the NIST/ARPA 1998 and 1999 Broadcast News evaluation[6], which included an optional ‘black-box’ Information Extraction–Named Entity (IE-NE) block, that further processed ASR system output.

Word Error Rate is obtained by first aligning the hypothesis generated by the ASR system with a reference in the evaluation database. This alignment problem is solved by recursively minimising the so called Levensthein distance[7] - a kind of string edit distance – that allows for insertions, deletions and substitutions in the hypothesised string to match it to the reference string. After the alignment the total amount of substitutions *S*, insertions *I* and deletions *D* is computed, and used in the following formula to compute the WER:

$$WER = \frac{I + D + S}{N} \qquad \text{Eq. 1}$$

where *N* is the total number of reference words.

This measurement is a sample of a random variable, and should be handled with care, if we are to establish properly the superiority of a given ASR system or component for the given evaluation task. Consequently, it’s important to use statistical methods to compare the measurements of different ASR systems. Such methods fall broadly in two classes, namely:

- Confidence intervals for the mean of the WER, which give a hint of how reliable is the measurement made.
- Statistical tests which try to ascertain, by means of hypothesis testing, which of the systems is the best.

1.4. Key questions of ASR evaluation

From the discussion above it should be clear that the key points in any evaluation are:

1. Task to evaluate on and its environment
2. Criteria and measures for evaluation
3. Evaluation data that is consistent with task, environment and criteria
4. Evaluation methodology

In the next point it’s explained how we handle each of the points above.

2. Evaluation framework

Our evaluation is divided in two phases, each with its own ASR task. A first one to validate the defined interfaces, and a second one to evaluate, using comparative technology assessment, the different feature extraction and acoustic modelling blocs of our project partners. Since the AURORA-2000 task/database was readily available to all the partners, and a robust – although not ‘realistic’ - recognition task too, it was decided to use it as standard database in the first phase, in which a lot of data and know-how must be exchanged, and it’s not critical to have ‘realistic’ data. In the second phase, however, our purpose is to evaluate in an in-car environment, and the intended application is a command-and-control one. That means that the task to evaluate on must certainly be a robust ASR task – in-car environments are noisy, with stationary and non-stationary noises -, and as command-and-control applications have rather small vocabularies, we thought that a good representative for such applications could be digit recognition. A suitable evaluation data for that task would be SpeechDat-Car, but it’s not available until next year.

As the output of our demonstrator ASR system is not further processed by any other LP system, natural selections for the criterion and measure were recognition performance and WER respectively. In the second phase, and since the demonstrator is on-line, processing speed will be added to the previous criterion. We decided to use the SCLITE package from NIST as our evaluation software, because it included a wide range of statistical tests, and is somehow becoming a kind of standard scoring package

As the aim of our evaluation is to identify the best feature extraction and/or acoustic modelling techniques, rather than the best ASR systems, in order to interface them to our on-line demonstrator, we were forced to impose stronger restrictions than the NIST/ARPA evaluation, since environment, task and task attributes – grammar, HMM model topology and lexicon – and even the implementation of the Viterbi decoder were fixed. Just the implementation of the feature extraction and the Acoustic Modelling blocks were allowed to vary. See figure for more details.

3. Results and discussion

Results on the AURORA-2000 task[8], i.e. for the first phase, are presented. This database contains two training sets, namely a multi-condition one, and a clean training with the same total number (8440) of training files. As for the tests sets, we have experimented with a reduced matched condition test set (testa) and reduced unmatched condition test set (testb). Both are composed of 16016 files divided in 16 sets of 1001 files each, one for each possible combination of noise (for testa the same noises as in multi-condition training, whereas for testb restaurant, street, airport and train station) and SNR (clean, 20, 10 and 5 dB).

In tables 1 and 3 the mean WER over all four noises – over 4x1001 files - is displayed, whereas tables 2 and 4 display the rank given to each of the recognisers – columns – by each of the statistical tests – rows – for each SNR. Statistical tests were performed pair wise, with 4x1001 files per test, i.e. the four noises have been included. Five different statistical test are included in the NIST SCLITE package, namely Analysis of Variance (anovar), Matched Pair Sentence (mapsswe), McNemar's (mcn), Sign and Wilcoxon (wilc) tests.

The DaimlerChrysler ASR[9] system uses Semi-Continuous HMMs (SCHMM) and a MFCC-based feature extraction with 12 coefficients plus on-line-normalised energy and Cepstral Mean Subtraction **(**CMS**).** Two baseline system configurations have been defined: one **3CBc** for clean-trained HMMs and another **LDAm** for noise-trained HMMs, where the *–m* and *–c* suffixes stand for multi-condition and clean. The former uses three different code-books – of 512, 256 and 256 classes respectively - to quantize CMS-MFCC static, delta and double-delta features, whereas the latter transforms a window of 9 static CMS-MFCC vectors with an LDA transform of (13x9)x32 dimensions to obtain a feature vector of 32 coefficients which is then quantized using a code-book of 127 classes, one for each state in the HMMs. There is a total of 13 HMMs, 11 for the digits, 1 one-state pause model and a 9-state noise model. Digit models have a different number of states, ranging from 8 to 15 states. No grammar was used during Viterbi decoding.

3.1. Results on testa (feature interface)

Three different feature extraction techniques are compared to our **LDAm** baseline feature extraction using the testa set, namely tandem multi-stream (**TDCm**) [10], Perceptual Linear Predictive (**PLPm**) and MFCC with CMS (**3CBm**) features. The first two have been generated using the SPRACHcore system and interfaced into our ASR system using the feature interface. The tandem approach adds the outputs of two MLPs – one for PLP and another of MSG features – and reduces the dimensionality of the combined vector to 24 coefficients using Principal Component Analysis (PCA), whereas PLP feature vector has 12 PLP coefficients plus on-line-normalised energy. All the models have been trained on the multi-condition set, as the suffix –m indicates. Results of the same tandem features with the SPRACHcore system are shown in the first column (**TANm**).

	TANm	TDCm	PLPm	LDAm	3CBm
SNR0	19.6	21,7	44,1	31,8	36,9
SNR10	2.4	2,4	6,8	6,5	7,8
SNR20	0.8	0,8	2,7	1,8	2,1
Clean	0.8	0,9	3,0	1,6	1,6

Tabel 1 Mean error rates over the four noises of the different feature extraction techniques on the testa set

Results of **TANm** are similar to those of **TDCm**, which means that the feature interface works. For all SNR and for all statistical tests **TDCm** is the best performing technique, whilst **LDAm** is the 2nd best ranked for all SNRs too, but for clean test data, where its performance is the same as **3CBm**. On the other hand, **PLPm** is the worst performing one, except for the 10dB SNR case, where it outperforms the **3CBm** technique.

		3CBm	LDAm	PLPm	TDCm
Clean	Anovar	2	2	3	1
	Mapsswe	2	2	3	1
	Mcn	2	2	3	1
	Sign	2	2	3	1
	Wilc	2	2	3	1
SNR20	Anovar	2	2	3	1
	Mapsswe	3	2	4	1
	Mcn	3	2	4	1
	Sign	2	2	3	1
	Wilc	3	2	4	1
SNR10	Anovar	2,5	2	2,5	1
	Mapsswe	3	2	2	1
	Mcn	3	2	2	1
	Sign	2,5	2	2,5	1
	Wilc	3	2	2	1
SNR0	Anovar	3	2	4	1
	Mapsswe	3	2	4	1
	Mcn	3	2	4	1
	Sign	3	2	4	1
	Wilc	3	2	4	1

Table 2 Ranking of each ASR technique according to the different statistical tests for each SNR in the testa set

3.2. Results on testb (state-likelihoods interface)

The testb set is used to evaluate the techniques trained using the clean training set and compare them to our multi-condition-trained baseline system. This is so because both the clean-trained and multi-condition-trained are in mismatched test conditions when tested on the testb, and therefore none of them has *a priori* advantage. As in the previous test set **LDAm** is our baseline trained on multi-condition train set, whereas **PLPc** and **3CBc** are exactly the same features of the previous example but with HMMs trained on clean speech. On the other hand **MDFc** stands for the Missing Data-Fuzzy Masks technique whose state-likelihoods have been generated by the CTK Toolkit of Sheffield University, and interfaced into our systems through the likelihoods interface. This technique uses a 32-filter gammatone filter-bank, plus delta coefficients, and feeds it into a Missing Data classificator [11], which uses the same HMM topology as described in 3, but Continuous Density HMMs (CDHMMs) with 7 gaussians per state. In the first two columns (**CTKc** and **CDCc**) we show the baseline results of the whole CTK system and with its state-likelihoods decoded with our ASR system too.

	CTKc	CDCc	MDFc	PLPc	LDAm	3CBc
SNR0	91,7	89,1	52,9	75,4	46,9	79,5
SNR10	76,1	76,5	15,2	21,9	10,3	27
SNR20	39,1	40,4	4,9	3,9	2	3,7
Clean	1,4	1,5	1,9	0,8	1,6	0,7

Tabel 3 Mean error rates over the four noises of the different ASR techniques on the testb set

Results of **CTKc** and **CDCc** are once again very similar, which demonstrates that the likelihoods interface works. For clean speech **PLPc** and 3CBc show the best performance, while **LDAm** and **MDFc** stay in the second position with nearly double WER. For high SNR **LDAm** is the best performing technique, followed by **3CBc** and **PLPc**. For low SNR **LDAm** is the best technique too, but **MDFc** outperforms both 3CBc and **PLPc**, and the latter the **3CBc**, which as expected is the worst performing in highly noisy conditions. At this point, it is interesting to note that even though **LDAm** has been trained using other noises than those in testb, it stills outperforms any other clean-trained technique in noisy conditions.

		3CBc	LDAm	PLPc	MDFc
Clean	Anovar	1	2	1	2
	Mapsswe	1	2	1	2
	Mcn	1	2	1	2
	Sign	1	2	1	2
	Wilc	1	2	1	2
SNR20	Anovar	2	1	2	2
	Mapsswe	2	1	2	3
	Mcn	2	1	2	3
	Sign	2	1	2	2
	Wilc	2	1	2	3
SNR10	Anovar	4	1	3	2
	Mapsswe	4	1	3	2
	Mcn	4	1	3	2
	Sign	4	1	3	2
	Wilc	4	1	3	2
SNR0	Anovar	3	1	3	2
	Mapsswe	4	1	3	2
	Mcn	3	1	3	2
	Sign	4	1	3	2
	Wilc	4	1	3	2

Table 4 Ranking of each ASR technique according to the different statistical tests for each SNR in the testb set

4. Conclusions

Terminology of ASR system evaluation has been overviewed in the introduction, stating which are the key points to consider while planning a successful evaluation framework. An analysis of our evaluation framework based on those key points has been done and discussed, with special emphasis on our 'evaluation methodology' based on interfaces between the evaluated ASR systems. Finally the results of the first phase of our evaluation – validation of interfaces - have been presented. As the results demonstrate, both the feature interface and the likelihood interface are successful. Further comparison of the results show that among the noise-trained techniques, the Tandem Multi-Stream approach scores far better than any of our baseline techniques, whereas PLP is clearly worse. In the clean-trained case, PLP and Missing Data Fuzzy Masks perform in low SNR conditions better than our clean-trained baseline, but are worse in noise than our noise-trained baseline, although it is tested in mismatched conditions too. To summarise, it seems to be better to train with noise, if test conditions are known to be noisy, even though the noises are different.

5. Acknowledgements

The authors would like to express their thankfulness to Viktoria Maier for her help in obtaining the results in section 3. This work was supported by EC RESPITE project and EC SPHEAR project.

6. References

[1] R.Crouch, R.Gaizauskas, K.Netter: *Report of the Study Group on Assessment and Evaluation*. April 1995

[2] R.Gaizauskas: *Evaluation in language and speech technology*. Computer Speech & Language, October 1998, Vol. 12, Nr 4

[3] NIST*: Benchmark tests for Spoken Language technology evaluations*. http://www.nist.gov/speech/ tests/index.htm

[4] G.A Miller, P.E. Nicely: *An analysis of perceptual confusions among some english consonants*. Journal of the ASA,Vol. 27, No. 2, March 1955.

[5] M. Boros,W. Eckert,F. Gallwitz,G. Görz,G. Hanrieder,H. Niemann: *Towards understanding spontaneous speech: word accuracy vs. concept accuracy*. Proc. ICSLP'96

[6] NIST: *1998 Hub-4 Broadcast News Evaluation* http://www.nist.gov/speech/tests/bnr/hub4_98/hub4_98.htm

[7] J. Picone, K.M. Goudie Marshall, G.R. Doddington, W. Fisher: *Automatic text alignment for speech system evaluation*. IEEE Trans. on ASSP. Vol. ASSP-34, No 4, August 1996.

[8] Hirsch H.G. and Pearce D. *The AURORA experimental framework for the performance evaluation of speech recognition systems under noisy conditions*. ISCA ITR Workshop ASR 2000: ASR: challenges for the next millenium

[9] F. Class, A. Kaltenmeier, and P. Regel-Brietzmann: *Optimization of an HMM-Based Continuous Speech Recognizer*. Proceedings Eurospeech, Berlin, 1993, 803-806.

[10] H. Hermansky, D. Ellis, S. Sharma: *Tandem Connectionist feature extraction for conventional HMM systems*. Proc. ICASSP 2000, Istambul.

[11] M.P. Cooke, P.D. Green, L. Josifovski and A. Vizinho: *Robust automatic speech recognition with missing and unreliable acoustic data*. Speech Communication, Vol. 34, No 3, June 2001.