Combining Forward-based and Backward-based Decoders for Improved Speech Recognition Performance

Denis Jouvet[1,2,3], *Dominique Fohr*[1,2,3]
Speech Group, LORIA
[1]Inria, Villers-lès-Nancy, F-54600, France
[2]Université de Lorraine, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
[3]CNRS, LORIA, UMR 7503, Villers-lès-Nancy, F-54600, France
{ denis.jouvet, dominique.fohr } @ loria.fr

Abstract

Combining outputs of speech recognizers is a known way of increasing speech recognition performance. The ROVER approach handles efficiently such combinations. In this paper we show that the best performance is not achieved by combining the outputs of the best set of recognizers, but rather by combining outputs of recognizers that rely on different processing components, and in particular on a different order (backward vs. forward) for processing speech frames. Indeed, much better speech recognition results were obtained by combining outputs of sphinx-based recognizers with outputs of Julius-based recognizers than by combining the same number of outputs from only sphinx-based recognizers, even if the individual sphinx-based systems led to better results than the individual Julius-based recognizers. Further experiments have also been conducted using sphinx-based tools for processing speech frames in reverse order (i.e. backward in time). The results clearly show that combining forward-based and backward-based decoders provide significant improvement with respect to a combination of forward only or backward only decoders. Experiments have been conducted on the ESTER2 and ETAPE speech corpora. Overall, combining sphinx-based and Julius-based systems led to 18.6% word error rate on ESTER2 test data, and 24.5% word error rate on ETAPE test data.

Index Terms: speech transcription, speech recognition, combining speech recognizer outputs, ROVER

1. Introduction

Speech transcription systems are getting more and more complex every year. They run in several passes, and typically start with a speaker segmentation and clustering process. Each audio segment is then classified according to speech quality (studio vs. telephone) and speaker gender. Finally each audio speech segment is decoded with the most adequate set of acoustic models. Extra passes may be applied for refining the decoding with adapted features and/or adapted models, or rescoring the hypotheses with more complex language models.

Besides elaborating more and more complex systems another approach for improving the speech transcription performance consists in combining several recognition systems. Combination possibilities span all the modules, from the acoustic input features, up to the combination of the recognition outputs, and going through combinations in the decoding process itself. All combinations rely on the idea that different systems (or modules) bring different pieces of information, and their combination should take benefit of the strengths of each of them.

A well-known combination approach consists in combining the outputs of different speech recognition systems through the ROVER procedure [1]. This procedure aligns the different hypotheses, and relies on a voting procedure for determining the best candidate word results. Good performances are obtained with such an approach, and it is also well known that the results of the best recognizer should be used for anchoring the alignment process. The procedure has then been enriched through the handling of a language model for helping the decision process [2] or through the introduction of a classifier for deciding on the best answer for each word using more detailed information than just the frequency of occurrences of the words and their confidence measures [3]. Other extensions involve dealing with the combination of confusion networks [4] instead of combining just the best hypothesis provided by each system. Tighter combinations of systems are also investigated to take benefit of several systems. This includes for example the exploitation of n-gram generated from the decoding with auxiliary systems for adjusting dynamically the language model used by the main decoder [5]. Some studies have also been conducted for running such processes with a low latency [6].

In this paper we are not concerned by complex systems, but by a detailed analysis of the combination of rather simple systems in order to understand what are the most important criteria when combining the output of several speech recognizers. Are the best results obtained by combining the best recognizers? Should other criteria be taken into account? And, another question that arises from the speech recognition systems used: does the different decoding directions, forward with respect to time frames in the standard sphinx system [7] and backward in the second pass (A*-based) of the Julius decoder [8], lead to complementary systems?

The paper is organized as follows. Section 2 describes the speech corpora that are used in this study. Section 3 presents the various transcription systems, based either on the sphinx toolkit or on the Julius toolkit, which were developed and used in the combinations. Section 4 compares and analyses various combinations of speech recognizer outputs through contrastive experiments. Finally a conclusion ends the paper.

2. Speech Corpora

The speech corpora used in the experiments come from the ESTER2 [9] and the ETAPE [10],[11] evaluation campaigns, and the EPAC [12],[13] project. The ESTER2 and EPAC data are French broadcast news collected from various radio channels, thus they contain prepared speech, plus interviews. A large part of the speech data is of studio quality, and some parts are of telephone quality. On the opposite, the ETAPE data correspond to debates collected from various radio and TV channels. Thus this is mainly spontaneous speech.

The acoustic models used later in the experiments are trained from the speech data of the ESTER2 and ETAPE train

sets, as well as the transcribed data from the EPAC corpus. The training data amounts to almost 300 hours of signal and almost 4 million running words.

The development and test sets of the ESTER2 and ETAPE data are used in the experiments reported below. The results (word error rates) are given for the non-African radios of the ESTER2 development set (named Dev-na in following tables - about 42,000 running words), for the non-African radios of the ESTER2 test set (named Test-na in following tables - about 63,000 running words) and for the whole ESTER2 test set (about 79,000 running words). Results are also provided for the whole ETAPE development set (about 82,000 running words) and for the whole ETAPE test set (about 82,000 running words). Performance evaluation on the ESTER2 data was carried on using the sclite tool [14] according to the ESTER2 campaign protocol. Performance on the ETAPE data was evaluated using the new LNE tools according to the ETAPE evaluation protocol [10] (and results are reported for non-overlapping speech data and for case independent mode).

3. Speech Recognition Systems

Several speech transcription systems are considered in the following experiments. They all start by applying a speaker segmentation and clustering process. Then each segment is classified according to its speech quality: studio quality or telephone quality. For the remaining processing, i.e. the speech recognition part, a set of systems rely on the sphinx decoder [7], while another set of systems rely on the Julius decoder [15],[8].

3.1. Sphinx-based systems

The standard sphinx-based transcription systems do a 2-pass decoding using a lexicon of about 95,000 words and a trigram language model. The pronunciation lexicons were obtained using the pronunciation variants present in the BDLEX [17] lexicon and in in-house pronunciation lexicons; then, for the remaining words, the pronunciation variants were obtained automatically using JMM-based and CRF-based Grapheme-to-Phoneme converters [18]. The trigram language model was trained using the SRILM tools [19] and various text corpora: more than 500 million words of newspaper data from 1987 to 2007; several million words from transcriptions of various radio broadcast shows; more than 800 million words from the French Gigaword corpus [20] from 1994 to 2008; plus 300 million words of web data collected in 2011 from various web sources, and thus mainly covering recent years. The language model weight for decoding (fudge factor) has been optimized on the development sets of the ESTER2 and ETAPE corpora, using the Sf.Va.Ms configuration (cf. description below).

For each system, a set of four acoustic models are used, corresponding respectively to the male and to the female adaptation of the studio quality and of the telephone quality models. Context-dependent phoneme units are used, for a total of 7,500 shared densities (senones), each of them having 64 Gaussian components. The first pass does a decoding of each audio segment using the most adequate acoustic model. The second decoding pass relies on a VTLN adaptation of the features and on a MLLR adaptation of the acoustic models.

Several systems have been used. They differ with respect to the choice of basic units, and the choice of acoustic features. Three sets of acoustic features have been considered: sphinx MFCC features (suffix .Ms), HTK [16] MFCC features (.Mh), and HTK PLP features (.Ph). In every case, the first 12 frame coefficients are used, plus the logarithm of the frame energy, and their first and second temporal derivatives.

Two sets of basic units have been considered. One set, noted .Va. (for *Vowels all*) uses all the phonemes defined in the BDLEX [17] pronunciation lexicon, whereas in the second set, noted .Vm. (for *Vowels merged*), the aperture of the vowels is not considered; hence we merge the open and the close /o/, the open and the close /e/, as well as the open and the close /ø/.

This led to the following sphinx-based speech transcription systems, which process frames in the standard time forward order; hence they are prefixed with Sf.:

Sf.Va.Ms: standard phoneme units, Sphinx MFCC features.
Sf.Va.Mh: standard phoneme units, HTK MFCC features.
Sf.Va.Ph: standard phoneme units, HTK PLP features.
Sf.Vm.Ms: merging vowel apertures, Sphinx MFCC features.
Sf.Vm.Mh: merging vowel apertures, HTK MFCC features.
Sf.Vm.Ph: merging vowel apertures, HTK PLP features.

Table 1: *Word error rates of sphinx-based forward systems*.

Sphinx forward system (Sf.)	ESTER2			ETAPE	
	Dev-na	Test-na	Test	Dev	Test
Sf.Va.Ms	21.3%	21.3%	23.1%	28.1%	29.4%
Sf.Va.Mh	**20.7%**	21.2%	22.9%	**27.7%**	**28.6%**
Sf.Va.Ph	21.3%	**21.0%**	**22.8%**	28.0%	28.9%
Sf.Vm.Ms	21.5%	21.8%	23.6%	28.7%	29.9%
Sf.Vm.Mh	21.6%	21.5%	23.2%	28.5%	29.3%
Sf.Vm.Ph	21.5%	21.6%	23.3%	28.6%	29.4%

The results in Table 1 show that the different sphinx-based forward systems have recognition performance in the same range. Performance on the non-African radios of the ESTER2 development and test sets are similar (columns Dev-na and Test-na). Performance on the ETAPE development and test sets are also rather close (about 1% difference). Overall, there is a slight advantage for using all the standard phonemes units (.Va.) and the HTK MFCC features (.Mh).

In order to investigate the impact of backward processing of the frames by the speech recognizer, the same set of systems have been developed in a backward processing approach: frames of each audio segment were given to the training tool and to the decoder in a reverse time order (i.e. last frame of each audio segment was given first). To have a consistent system, the pronunciation of each word in the lexicon was also reversed, and language models were re-estimated after reversing all the text sentences. The corresponding systems are used in section 4.1, and they are referred to by prefix Sb. (for Sphinx backward). They achieved similar performance as the forward based systems. Detailed performance of each individual system is not reported here. However, Table 6, shows that the combination of the 6 sphinx-based backward systems (3.Sb.Va+3.Sb.Vm) provides results which are almost identical to the combination of the 6 sphinx-based forward systems (3.Sf.Va+3.Sf.Vm).

3.2. Julius-based systems

The second set of transcription systems rely on the Julius decoder. They also use acoustic models dependent on the quality of the signal: studio quality vs. telephone quality. The acoustic models were developed using the HTK toolkit [16]. Context-dependent phoneme units are used, they are modeled with 6,000 shared states/densities, and each mixture density

has 62 Gaussian components. An HLDA transformation is applied on the acoustic features before modeling. These transcription systems run also in two passes. The second transcription pass relies on SAT adapted models.

For these systems the lexicon has a similar size, about 96,000 entries. As for the sphinx pronunciation lexicon, the pronunciation variants were first extracted from the BDLEX and other available in-house pronunciation lexicons, then for the remaining words, which are assumed to be mainly proper names, the pronunciations variants were obtained using a CRF-based Grapheme-to-Phoneme converter specialized for proper names.

It is also important to mention that the Julius decoder runs in a forward-backward mode. The forward pass relies on a bigram and generates a search graph; then, the backward A* pass explores this graph guided by a 4-gram language model. The forward bigram and backward 4-gram models were trained using the SRILM toolkit and the same text data as described before.

Several systems have been developed. They differ with respect to the choice of basic phonetic units and the choice of acoustic features. Two sets of acoustic features have been considered: HTK MFCC features (.Mh) and HTK PLP features (.Ph); 12 coefficients plus logarithm of frame energy are used, and an HLDA transformation is applied on a 9 frame window to provide the 40 input modeling coefficients. As previously, two sets of phoneme units have been considered, by taking into account (.Va.) or ignoring (.Vm.) the aperture of the vowels.

This led to the following Julius-based speech transcription systems:

Jb.Va.Mh: standard phoneme units, HTK MFCC features.
Jb.Vm.Mh: merging vowel apertures, HTK MFCC features.
Jb.Vm.Ph: merging vowel apertures, HTK PLP features.

Table 2: *Word error rates of Julius-based systems*.

Julius backward system (Jb.)	ESTER2			ETAPE	
	Dev-na	Test-na	Test	Dev	Test
Jb.Va.Mh	26.2%	25.3%	28.1%	32.2%	32.0%
Jb.Vm.Mh	**23.5%**	**23.5%**	**26.6%**	**30.1%**	**30.6%**
Jb.Vm.Ph	26.5%	25.6%	29.0%	32.7%	32.5%

The results in Table 2 exhibit rather similar word error rates on the non-African radios of the ESTER2 development and test data, as well as between the ETAPE development and test sets. Some parameters of the Julius decoder (such as the language model weights and word insertion penalty) have been optimized on the ETAPE development data using the Jb.Vm.Mh configuration (HTK MFCC features and merging of vowel apertures) and these parameters have been used for all configurations. This might explain the best results obtained by this Jb.Vm.Mh configuration compared to the other Julius-based configurations.

4. Combining Speech Recognizer Outputs

This section investigates the combination of the outputs of the recognizers using the ROVER [1] approach. Confidence measures are not used in the ROVER combinations. The first sub-section presents combinations of the standard forward sphinx-based recognition system outputs with Julius-based recognition system outputs. The second sub-section focused on comparing various combinations of sphinx-based recognition system outputs to get an insight on the benefit of combining forward-based and backward-based decoders.

4.1. Combining sphinx- and Julius-based systems

The first set of combinations explored involves only 3 transcription systems each:

1.Sf.Va+2.**Sf**.Va: Sf.Va.Ms & Sf.Va.Mh & Sf.Va.Ph
1.Sf.Va+2.**Sf**.Vm: Sf.Va.Ms & Sf.Vm.Mh & Sf.Vm.Ph
1.Sf.Va+2.**Jb**.Vm: Sf.Va.Ms & Jb.Vm.Mh & Jb.Vm.Ph

Table 3: *Word error rates of ROVER combinations*.

Rover combination (3 ASR systems)	ESTER2			ETAPE	
	Dev-na	Test-na	Test	Dev	Test
1.Sf.Va+2.**Sf**.Va	**19.5%**	**19.6%**	**21.3%**	**26.3%**	**27.3%**
1.Sf.Va+2.**Sf**.Vm	19.9%	19.7%	**21.3%**	26.7%	27.7%
1.Sf.Va+2.**Jb**.Vm	20.1%	20.2%	22.3%	25.7%	26.7%

As shown by the results reported in Table 3, combining the outputs of three systems leads to significant performance improvement on every data set (1.5 to 2% absolute word error rate reduction compared to the individual system results reported in Table 1). Large word error rate reductions are observed in all cases, whether we combine only sphinx-based systems or sphinx-based with Julius-based systems.

The next set of experiments involves combinations of more system outputs: three sphinx-based systems (corresponding to the 1.Sf.Va+2.**Sf**.Va combination above), plus two additional systems; on the one side two other sphinx-based systems, and on the other side two Julius-based systems:

3.**Sf**.Va+2.**Sf**.Vm: Sf.Va.Ms & Sf.Va.Mh & Sf.Va.Ph
& Sf.Vm.Mh & Sf.Vm.Ph
3.**Sf**.Va+2.**Jb**.Vm: Sf.Va.Ms & Sf.Va.Mh & Sf.Va.Ph
& Jb.Vm.Mh & Jb.Vm.Ph

Table 4: *Word error rates of ROVER combinations of sphinx-based and Julius-based systems*.

Rover combination (5 ASR systems)	ESTER2			ETAPE	
	Dev-na	Test-na	Test	Dev	Test
3.**Sf**.Va+2.**Sf**.Vm	19.0%	19.0%	20.5%	25.8%	26.7%
3.**Sf**.Va+2.**Jb**.Vm	**17.6%**	**17.7%**	**19.2%**	**24.1%**	**24.9%**

Table 4 shows the word error rates obtained when combining the outputs of five systems. The results display a large difference between the two combinations. The combination of two Julius-based systems with three sphinx-based systems (i.e. 3.Sf.Va+2.Jb.Vm) leads to word error rates that are about 1.5% absolute lower than those resulting from the combination of 5 sphinx systems (3.Sf.Va+2.Sf.Vm). The difference between the two combinations is limited to the two last recognizers involved. These four systems (2.Sf.Vm and 2.Jb.Vm) involve the same set of phonetic units (Wm, i.e. after merging of vowel apertures) and the same input features: HTK MFCC (for Sf.Vm.Mh and Jb.Vm.Mh) and HTK PLP (for Sf.Vm.Ph and Jb.Vm.Ph). It is also very important to note that although the Julius-based recognizers alone provide significantly worse results than the corresponding sphinx-based recognizers (cf. Table2 and Table 1: Sf.Vm.Mh performance better than Jb.Vm.Mh performance, and Sf.Vm.Ph performance better than Jb.Vm.Ph performance), their combination with other Sphinx-based recognizers leads to a much larger reduction in the word error rates.

This experiment confirms that it is crucial and better to combine systems that are complementary than to combine a set of systems which are individually better but less complementary. One possible explanation of this behavior might be correlated to the operating mode of the recognizers. The sphinx decoder processes the speech signal in a single forward pass, whereas the Julius decoder deals with the speech signal in a forward plus a backward process. The forward pass of the sphinx decoder and the backward last pass of the Julius decoder should normally lead to different search spaces, and one might expect that one system is likely to recover errors made by the other system.

The benefit of combining systems is also demonstrated by these last experiments of system combinations:

5.**Sf**.V*+2.**Jb**.**Vm**: Sf.Va.Ms & Sf.Va.Mh & Sf.Va.Ph
& Sf.Vm.Mh & Sf.Vm.Ph
& Jb.Vm.Mh & Jb.Vm.Ph

5.**Sf**.V*+3.**Jb**.**V***: Sf.Va.Ms & Sf.Va.Mh & Sf.Va.Ph
& Sf.Vm.Mh & Sf.Vm.Ph
& Jb.Vm.Mh & Jb.Vm.Ph & Jb.Va.Mh

Table 5: *Word error rates of ROVER combinations of sphinx-based and Julius-based systems*.

Rover combination (Sphinx & Julius ASR)	ESTER2			ETAPE	
	Dev-na	Test-na	Test	Dev	Test
5.**Sf**.V*+2.**Jb**.**Vm**	17.7%	17.8%	19.2%	24.4%	25.1%
5.**Sf**.V*+3.**Jb**.**V***	**17.3%**	**17.3%**	**18.6%**	**23.7%**	**24.5%**

Although the first combination (5.Sf.V*+2.Jb.Vm) involves more transcription systems than the previous one (3.Sf.Va+2.Jb.Vm), it does not provide better performance. This might be due to the largely unbalanced set of systems: five sphinx-based systems against only two Julius-based systems. Such unbalanced distribution of the systems penalized the less represented ones in the voting process of the ROVER procedure. Adding an extra Julius-based system restores the balance and improves the results. These experiments clearly show that combining several systems based on different features, different phonetic units and different decoding engines leads to large reductions in the word error rates; results hold on the different data subsets corresponding to broadcast news transcription (ESTER2 task) and TV and radio debates transcription (ETAPE task).

4.2. Combining forward- and backward-based sphinx systems

These new set of combinations are intended to investigate the benefit of combining forward-based and backward-based decoding outputs. In order to get the best possible insight on this phenomenon, four combinations of six systems each are compared. Two combinations involve only forward-based (Sf.) or backward-based (Sb.) sphinx systems, and the two other combinations involve a mixed of approaches (three forward-based and three backward-based):

3.**Sf**.Va+3.**Sf**.Vm: Sf.Va.Ms & Sf.Va.Mh & Sf.Va.Ph
& Sf.Vm.Ms & Sf.Vm.Mh & Sf.Vm.Ph

3.S**b**.Va+3.S**b**.Vm: Sb.Va.Ms & Sb.Va.Mh & Sb.Va.Ph
& Sb.Vm.Ms & Sb.Vm.Mh & Sb.Vm.Ph

3.**Sf**.Va+3.S**b**.Vm: Sf.Va.Ms & Sf.Va.Mh & Sf.Va.Ph
& Sb.Vm.Ms & Sb.Vm.Mh & Sb.Vm.Ph

3.S**b**.Va+3.**Sf**.Vm: Sb.Va.Ms & Sb.Va.Mh & Sb.Va.Ph
& Sf.Vm.Ms & Sf.Vm.Mh & Sf.Vm.Ph

Table 6: *Word error rates of ROVER combinations of forward-based and backward-based sphinx systems*.

Rover combination (6 Sphinx systems)	ESTER2			ETAPE	
	Dev-na	Test-na	Test	Dev	Test
3.**Sf**.Va+3.**Sf**.Vm	18.9%	19.0%	20.5%	25.7%	26.6%
3.S**b**.Va+3.S**b**.Vm	19.1%	19.0%	20.3%	25.7%	26.5%
3.**Sf**.Va+3.S**b**.Vm	**18.0%**	**18.2%**	**19.5%**	**24.7%**	**25.4%**
3.S**b**.Va+3.**Sf**.Vm	18.1%	18.3%	19.6%	24.8%	25.5%

Results in Table 6 show that the combination of forward-based sphinx systems (3.Sf.Va+3.Sf.Vm) and the combination of backward-based sphinx systems (3.Sb.Va+3.Sb.Vm) lead to very similar recognition performance. The two other combinations of three forward-based and three backward-based systems (3.Sf.Va+3.Sb.Vm and 3.Sb.Va+3.Sf.Vm) also lead to very similar recognition performance.

It is important to note that every combination involves two systems with sphinx MFCC features (.Ms), two with HTK MFCC features (.Mh), and two with HTK PLP features (.Ph). With respect to the phone units, in each combination three systems use the standard phoneme units (.Va.) and three systems use the reduced set resulting from the merging of vowels apertures (.Vm.). Hence the only difference between the various combinations is the fact that some systems are forward-based (Sf.) whether some others are backward-based (Sb.). Comparing the last two lines (3.S**f**.Va+3.S**b**.Vm and 3.S**b**.Va+3.S**f**.Vm) to the first two lines (3.S**f**.Va+3.S**f**.Vm and 3.S**b**.Va+3.S**b**.Vm) shows that the combination of forward-based and backward-based systems provides much better results than the combination of forward only or backward only systems. On average, there is a 0.7% to 1.0% absolute error rate reduction due to the simple fact of combining forward-based and backward-based systems.

5. Conclusion

This paper has investigated the combination of several transcription systems in order to better understand what is important in the combination of the output of recognizers with the ROVER procedure.

Several transcription systems were developed which differ one from the other with respect to the acoustic features, the set of phonetic units, and also the decoding engine. The experiments clearly shows that it is much more efficient to combine outputs of recognizers that behave differently, even if they do not have the best performance, rather than selecting only the best performing systems (which might have very similar behavior). For the ROVER procedure to be efficient, it is also important to have a balanced set of systems, otherwise the less represented ones might be useless because of the voting process of the ROVER procedure.

To get a better insight on the behavior of system combinations, a specific set of experiments was conducted, and showed that the combination of forward-based and backward-based decoders leads to much better results (0.7% to 1.0% absolute error rate reduction) than the combination of forward only or backward only decoders. Further analysis is needed to compare the errors made by backward-based systems to those of the forward-based systems; and possibly derive new combination approaches of such systems.

Overall, the combination of several simple systems leads to state of the art performance on the ESTER2 and on the ETAPE transcription tasks.

6. References

[1] J.G. Fiscus, “A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)”, in *Proc. ASRU’1997, IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 347-354, Dec. 1997.

[2] H. Schwenk, and J.-L. Gauvain, “Combining multiple speech recognizers using voting and language model information”, in *Proc. INTERSPEECH’2000*, pp. 915-918, 2000.

[3] D. Hillard, B. Hoffmeister, M. Ostendorf, R. Schlüter, and H. Ney, “iROVER: improving system combination with classification”, in *Conf. of the North American Chapter of the Association for Computational Linguistics*, Rochester, New-York, pp.65-68, April 2007.

[4] G. Evermann, and P.C. Woodland, “Posterior probability decoding, confidence estimation and system combination”, in *Proc. NIST Speech Transcription Workshop*, 2000.

[5] F. Bougares, Y. Estève, P. Deléglise, G. Linares, "Bag of n-gram driven decoding for LVCSR system harnessing", in *Proc. ASRU’2011, IEEE Workshop on Automatic Speech Recognition and Understanding*, Hawaï, USA, Dec. 2011.

[6] F. Bougares, Y. Estève, P. Deléglise, G. Linares, "Low latency combination of parallelized single-pass LVCSR", in *Proc. INTERSPEECH’2012*, Portland, USA, Sept. 2012.

[7] Sphinx. [Online]: http://cmusphinx.sourceforge.net/, 2011.

[8] Julius. [Online]: http://julius.sourceforge.jp/en_index.php

[9] S. Galliano, G. Gravier, and L. Chaubard, “The Ester 2 evaluation campaign for rich transcription of French broadcasts”, in *Proc. INTERSPEECH'2009, 10th Annual Conf. of the Int. Speech Communication Association*, Brighton, UK, pp. 2583-2586, Sept. 2009.

[10] G. Gravier, and G. Adda, *Evaluations en traitement automatique de la parole (ETAPE), Evaluation Plan*, Etape 2011, version 2.0, 2011.

[11] G. Gravier, G. Adda, N. Paulsson, M. Carré, A. Giraudel, and O. Galibert, "The ETAPE corpus for the evaluation of speech-based TV content processing in the French language", in *Proc. LREC’2012, Int. Conf. on Language Resources, Evaluation and Corpora*, Istanbul, Turkey, May 2012.

[12] Y. Estève, T. Bazillon, J.-Y. Antoine, F. Béchet, and J. Farinas, "The EPAC corpus: Manual and automatic annotations of conversational speech in French broadcast news", in *Proc. LREC’2010, European Conf. on Language Resources and Evaluation*, Valetta, Malta, May 2010.

[13] Corpus EPAC: Transcriptions orthographiques, catalogue ELRA (http://catalog.elra.info), reference ELRA-S0305.

[14] NIST evaluation tools: http://www.itl.nist.gov/iad/mig//tools/

[15] A. Lee, and T. Kawahara, "Recent Development of Open-Source Speech Recognition Engine Julius", in *Proc. APSIPA ASC'2009, Asia-Pacific Signal and Information Processing Association Annual Summit and Conf.*, Sapporo, Japan, 2009.

[16] HTK. [Online]: http://htk.eng.cam.ac.uk/

[17] M. de Calmès, and G. Pérennou, “BDLEX : a Lexicon for Spoken and Written French.” in *Proc. LREC’1998, 1st Int. Conf. on Language Resources & Evaluation*, pp.1129-1136, Grenade. 1998.

[18] I. Illina, D. Fohr, and D. Jouvet, “Grapheme-to-Phoneme Conversion using Conditional Random Fields”, in *Proc. INTERSPEECH'2011*, Florence, Italy, Aug. 2011.

[19] A. Stolcke, “SRILM - An Extensible Language Modeling Toolkit”, in *Proc. ICSLP’2002, Int. Conf. on Spoken Language Processing*, Denver, Colorado, Sept. 2002.

[20] A. Mendonça, D. Graff, and D. DiPersio, “French Gigaword Second Edition”, Linguistic Data Consortium, Philadelphia, 2009.