Integrating imperfect transcripts into speech recognition systems for building high-quality corpora

Benjamin Lecouteux, Georges Linares, Stanislas Oger

Integrating imperfect transcripts into speech recognition systems for building high-quality corpora

Benjamin Lecouteux, Georges Linarès, Stanislas Oger

Abstract

The training of state-of-the-art automatic speech recognition (ASR) systems requires huge relevant training corpora. The cost of such databases is high and remains a major limitation for the development of speech-enabled applications in particular contexts (e.g. low-density languages, or specialized domains). On the other hand, a large amount of data can be found in news prompts, movie subtitles or scripts, etc. The use of such data as training corpus could provide a low-cost solution to the acoustic model estimation problem. Unfortunately, prior transcripts are seldom exact with respect to the content of the speech signal, and suffer from a lack of temporal information. This paper tackles the issue of prompt-based speech corpora improvement, by addressing the problems mentioned above. We propose a method allowing to locate accurate transcript segments in speech signals and automatically correct errors or lack of transcript surrounding these segments. This method relies on a new decoding strategy where the search algorithm is driven by the imperfect transcription of the input utterances. The experiments are conducted on the French language, by using the ESTER database and a set of records (and associated prompts) from RTBF (Radio Télévision Belge Francophone). The results demonstrate the effectiveness of the proposed approach, in terms of both error correction and text-to-speech alignment.

Keywords: Speech processing, acoustic model training, text-to-speech alignment

1. Introduction

Recent evaluation campaigns have demonstrated the significant advances of speech recognition technologies, especially in handling adverse acoustic conditions and various speaking styles (spontaneous and/or interactive speech, unconstrained dialogues, etc., [12]). These improvements were obtained by the integration of new sophisticated techniques, such as system combination, accurate speaker adaptation, discriminative training, etc. Most state-of-the-art recognition systems rely on the Hidden Markov Model-based framework for acoustic and n-gram language modeling: the size and the relevance of training corpora remains a key issue in estimating such models. Moreover, recent techniques tend to increase the sensitivity of the acoustic models to their training corpus, especially discriminative training methods that involve decoding large and accurate training sets [51]. Therefore, a particular attention is paid to the quality and relevance of these resources; the acoustic models are frequently estimated on hundreds of hours of annotated speech. Unfortunately, manual annotation is very expensive; this constitutes a technical limit to the development of speech technology on specialized domains or low-density languages, where the economical interest is low.

On the other hand, some recordings are composed of speech produced following *a priori* written text. This is the case for movie scripts, political speeches, news prompts, etc. Transcripts may also be produced *a posteriori*, when speech data is archived, and/or indexed. The use of these data as training corpus could provide a low-cost way of extracting training corpora.

Building prompt-based training corpora faces two main difficulties. First, the speakers do not follow systematically their written source and the resulting utterance may be only an approximate transcription. This limits the interest of such transcripts for acoustic model training. Second, prompts usually suffer from a lack of temporal information, especially on large shows where news, talks and live segments (such as interviews) or reported speech alternate. In this case, prompted segments have to be found in a relatively large amount of speech data, without the time marks that are required by the training process.

This paper addresses these different issues of using imperfect transcripts, composed of prompts or close-captions, to build high-quality corpora. We propose a generic approach where the recognition engine is driven by imperfect transcripts. This method first spots *transcript islands* in the speech signal and then drives the ASR system to take advantage of these accurate

partial transcriptions. We call *transcript islands* small segments of transcripts that accurately match the pronounced speech, that are available prior to transcription, and for which precise timing is not known. The word *island* is used in order to emphasize the fact that these transcripts are generally scarce compared to the large quantity of unavailable or inaccurate transcripts.

The paper is structured as follows: Section 2 presents an overview of the techniques which are reported in the scientific literature on this field. In Section 3, we describe our approach for alignment and correction of imperfect transcripts. We propose Driven Decoding Algorithm (DDA) that integrates user-provided close-captions in the recognition engine. In Section 4, we tackle the issue of partially-prompted shows. We propose a *transcript island* spotting method that detects signal segments that match an *a priori* available partial transcripts. This method combined with the DDA framework allows us to extract clean speech corpora from partially-prompted speech databases, results are presented in section 5. Finally, Section 6 concludes the paper and proposes some perspectives.

2. Related work

2.1. Close-captions and the fidelity of prompts

The possibility of using prompts as corpus has been considered for a long time in the speech processing community. In [4], the major issues in using news prompts are studied; the authors note that the prompts are not always strictly read, sentences are forgotten or inserted and other sentences are pronounced with some word variations, and close-captions have to respect additional constraints due to the size of the screen where the caption is displayed. [39] evaluate a close-captions Word Error Rate (WER) of 10% to 20%. It is important to note that in the two cases (prompts or closed captions), the grammaticality of the transcripts remains correct, as they are produced beforehand by journalists or by transcriptionists.

The differences between the expected utterance and the actual spoken content cause several problems to acoustic model training; first, the alignment algorithm may be dramatically affected by errors; second, it is crucial to be confident in the transcript for estimating the acoustic models. In the next sections we present an overview of speech-to-text alignment methods on both correct and imperfect transcripts.

2.2. Speech-to-text alignment

Speech-to-text alignment is a well-studied issue in the speech processing field. The classical approach consists in modeling the word utterance by a graph of Hidden Markov Models (HMMs) corresponding to the transcription pronunciation graph. The alignment is performed by using the well-known Viterbi algorithm, which is effective on reasonably correct transcripts. Alternative approaches must be used in order to address the issue of the (highly) imperfect transcripts.

In [31], P.J. Moreno *et al.* evaluate a method aiming to align long audio documents with their exact transcripts in the context of automatic indexing of multimedia documents. This method relies on a search of synchronization points between the exact transcript and an ASR-provided transcription. The first step consists in estimating a language model on the exact transcript. Synchronized areas are isolated from the extracted segments with a match between the *a priori* and automatic transcripts. Documents are then segmented according to these small confidence islands; on each segment, a specific language model is estimated. The algorithm is run recursively on unaligned segments, until the convergence point is reached. This method is restricted to exact transcripts, but the reported experimental results demonstrated its effectiveness, 99% of the words are correctly aligned on a broadcast news task. Moreover, the effectiveness of this approach depends on the ASR system accuracy – which has to be good enough, and relies on the availability of correct transcripts.

Another strategy consists in generating the alignment that minimizes the edit distance T_{edit} between available transcripts and the outputs of an ASR system [4]. In [30], the synchronization of the recognition hypothesis T_{ASR} and the corresponding closed caption T_{CC} is performed by a transducer-based optimization process, where T_{aligns} is the result of the alignment between T_{ASR} and T_{CC}, and $\oplus$ is the composition operator:

$$T_{aligns} = T_{ASR} \oplus T_{edit} \oplus T_{cc} \tag{1}$$

The best path is computed by performing a best path search, by using the classical Dijkstra algorithm [10]. This operation is performed for each potential closed caption, the alignment hypotheses being validated by a simple decision rule. This method is evaluated for acoustic model training, by collecting the detected segments in a training corpus. Successive decoding passes must be performed and the alignment is computationally expensive.

In [50], Witbrock *et al.* use temporal information and align the closed captions to audio signals by using a classical dynamic time warping (DTW) algorithm [47, 20]. In [39] Placeway and Lafferty propose an integrated method, based on a translation model that maps the captions to appropriate word sequences. The translation model is a Markov chain where transitions represent deletions, insertions or substitutions. This method was implemented in the Sphinx III decoder [38], where caption segments are first timestamped in a first pass.

All these methods are ASR-based and may be dramatically influenced by high word error rates. In [49], the authors propose an approach based on a dictionary that, instead of words, contains phoneme sequences. This method allows one to deal with out-of-vocabulary words. In [14] Haubold *et al.* present a sub-lexical approach for the alignment of speech to highly imperfect transcripts.

2.3. Transcript island recognition and synchronization

The task of finding *transcript islands* is related to, but different from classical speech-to-text alignment. It consists in finding, in a set of transcripts, the part that matches mostly the current speech segment.

For example, in the case of prompt databases, large quantities of texts are available without timestamps. A *transcript island* recognition process allows one to select matching parts related to the content of speech segments.

In [4], the authors propose a method that selects segments from a large database, competing segments being chosen according to a simple two-string matching score. Some algorithms focus on spotting short *transcript islands*. In [43], Smith and Waterman propose to perform local sequence alignments by searching for matching areas, similarly to the techniques used for comparing two nucleotides or protein sequences. This algorithm compares segments of all possible lengths and offsets.

The problem of evaluating the similarity between a partial recognition hypothesis and a prior transcript segment may be viewed as an information retrieval issue. In text retrieval, the problem is to find documents that meet the user information needs, expressed as a query. Most of the approaches to this issue rely on the vector space model, where documents – and queries, are represented in a space D where the dimensions are the words that compose the documents. Words are extracted from the documents after stripping off stop words and stemming them [15]. The TF $\times$ IDF (Term Frequency $\times$ Inverse Document Frequency) measure is frequently used to get an estimation

of the information carried by a word [42, 41]. The similarities between queries and documents are computed, for example by using the cosine metric, and documents are ranked accordingly. This operation is fast and tractable in large documents. In [16], the authors propose a retrieval method based on clustering. It consists in splitting the text into small parts, with a confidence measure assigned to each part. The intersection between a cluster and query is used as an index of document relevance.

2.4. Unsupervised training of ASR systems

In [19, 48], the authors propose a study of unsupervised procedures for the training of acoustic models. Confidence measures are used to restrict unsupervised training to the words that are probably correct. Using only a few minutes of manually transcribed acoustic training data and an iterative process, initial models are improved. Moreover, in [19] the procedure is improved by training more than one recognizer and combining the recognition results with a ROVER. However, in [48] the final WER on the testing sets increased by 14% and 18% relative in comparison with a system trained on the full manual transcriptions. Generally, standard unsupervised training is not efficient for discriminative training which is highly sensitive to the transcription quality. However, in [52], the authors show that using a small amount of directed manual transcriptions (i.e. the proposed strategy is to incorporate some supervised data for the poorly recognized genre) is able to improve unsupervised discriminative training.

2.5. Approximated transcripts and ASR systems

As previously discussed, the need for large speech databases for acoustic model training is mainly due to the acoustic model learning paradigm: extensive HMM-based modeling requires the estimation of a large number of free parameters (classically, more than 10 million), requiring a large amount of training data (classically, hundreds of hours of annotated speech). Considering the low quality of closed caption-based corpora, most of the proposed approaches consist in using alternative strategies for model training, where both quality and quantity of training data are considered as macro-parameters of the estimation algorithm. In the next subsections, we present related work exploiting approximated transcripts.

2.5.1. Biased language models

The linguistic variability can be reduced, thanks to the contribution of an exact or imperfect transcript. The reduction of the overall linguistic space helps improving the ASR performances. This can be achieved by estimating a language model on the transcript itself. However, such a language model would probably be too specific when the speaker deviates from the original transcript. Therefore, this model is interpolated with a generic language model, following the well-known linear combination of language models (LMs) [18]. The approach is limited by the transcript quality and the interpolation weight should be dynamic. Moreover, experiments using only interpolation carried out in [39] show the limits of this approach: when the interpolation weight is small, the resulting language model does not strongly improve the transcription, and with a large weight, the models match excessively the specialized data.

In [9, 17, 28] the authors present two techniques for language model adaptation: the first uses a mixture-based model where mixing weights are computed using a first pass decoding, and the second is based on a cache containing words hypothesized in the past. In the two cases, a specific information can be integrated into the language model.

The cache model introduced by [21] proposes to increase the probability of the words that have occurred recently. The assumption behind this model is that if a word is used in a context, then this word is likely to be used again. The trigger model, as explored by [32] is a generalization of cache models: the long-distance dependency problem is addressed by interpolating n-grams with trigger pairs selected according to their mutual information.

However, cache models lack robustness, because boosted words are depending on the current hypothesis; an error can easily spread and too many hypotheses can be made with all trigger pairs: they are not easy to tune.

2.5.2. Prompts as training corpus for ASR

Some papers propose to perform ASR on the speech signal and to extract, from the output, the areas that match the prior transcript. In [50], the authors apply this principle to teletext / closed caption materials, for a re-estimation of the acoustic models, based on television input.

In [22], Lamel *et al.* propose a slightly supervised method for acoustic model training by using low-quality transcribed databases. This approach consists of three steps:

- automatically transcribe the training database.
- search for ASR output segments that match a part of the prior transcript.
- use the matching segments for acoustic model re-estimation.

The ASR output is used to find matching close captions. These segments are then used to re-estimate the acoustic models.

This idea is extended in [7], where closed captions are used in order to refine acoustic models. The authors use a confusion network (CN) to find similar parts between closed captions and audio signal: the ASR system generates a CN for each audio segment. Then, closed captions are aligned with the CN in order to extract matching parts. A threshold is used to cut off false assumptions. This method extracts more aligned data than the slightly supervised method [22], by using the entire ASR hypothesis.

In [5], this idea is applied jointly to Maximum Mutual Information Estimation (MMIE) or Minimum Phone Error (MPE) discriminative training.

In [33], the authors propose a light supervision method to acquire acoustic training data from speech associated to corresponding prompts. They estimate an interpolated language model using imperfect transcripts. The ASR output is aligned to the approximate transcripts and only matching words are selected for acoustic training. They yield 13% relative error rate reduction with 702 hours added to the baseline (141 hours of training data).

In [34, 35], the authors deals with multiple audio streams and translation systems in order to improve the ASR transcripts. In the case of the European Parliament, debates are translated into multiple languages. The multiple knowledge sources (Final Text Editions (FTE), audio signal) are used to supervise the primary system:

- FTE into the language model training data.
- Automatic 1000-best translation hypotheses for each language are also used for language model training data.
- Acoustic models are trained according to a general FTE supervision.

In [11], the authors propose a transcription alignment method using improved pronunciation models and HMM based transcripts, in order to reduce typical errors in the transcriptions. HMMs introduce more flexibility in the

transcripts: instead of aligning the plain transcription, a HMM is generated for each utterance allowing multiple pronunciations, multi-words and optional silences or noises. The Flexible Training Alignment applied to the Switchboard and CallHome corpora reduces significantly the WER. These results demonstrate how important the transcript quality is.

Another approach presented in [36, 37] corrects *a posteriori* ASR output using words in associated medical reports. Authors introduce a flexible automatic phonetic transcription to solve the issues of formatted entity prompts and alternative pronunciations. The methods based on phonetic similarity matching and text alignment on multiple levels of segmentation allow one to improve the accuracy of the transcriptions.

Another approach proposed by Placeway *et al.* [39] is to estimate a language model with prompts or closed captions. The estimated model is interpolated with a generic language model in the ASR system. This technique improves the results, but the information brought by subtitles is mostly drowned in the data quantity. Placeway *et al.* include a closed caption model into a beam search: the words that match the imperfect transcript are favored.

All related work allows exploiting imperfect transcripts in order to improve an ASR system. [36, 37, 11] are dedicated to *good* quality transcripts and do not influence the ASR system itself. The methods presented in [50, 22, 7, 6, 34, 35] are more classical approaches where additional data are used in order to improve/bias language models and acoustic models. Finally, the framework proposed by [39] directly introduces approximated transcripts into the ASR engine.

In our experiments, we mainly use two methods as baseline: the first one is the method proposed by [39], allowing to include prompts into a synchronous Viterbi decoding framework. The second baseline method will be the slightly supervised approach [22] which is widely used in the speech recognition community.

3. Error correction by Driven Decoding Algorithm

3.1. Background

Our goal is to exploit imperfect transcripts within an asynchronous decoder based on the A* algorithm, in the framework of a broadcast news system. We expect an **integrated approach** that does not require multiple passes.

We first present the characteristics of the *Laboratoire Informatique d'Avignon* (LIA) decoder. We also show how the information from imperfect transcripts can be integrated in the decoding process.

Then we evaluate a naive method that consists in combining a generic language model and a language model estimated on the imperfect transcripts.

Finally, we propose a new approach that relies on driving the ASR system search algorithm with the imperfect transcripts.

3.2. The LIA French broadcast news system

The LIA Broadcast News system relies on the Speeral decoder [27] and the Alize-based segmenter [2]: segments are automatically segmented by speaker (in order to adapt the acoustic models by using Maximum Likelihood Linear Regression method) and their size is limited to 30 seconds.

Here, we use the system involved in the ESTER evaluation campaign [13]. Cross-word context-dependent acoustic models with 230k Gaussians are used. Tying is achieved by decision trees, by using acoustic context related questions. We train the acoustic models on ESTER materials (about 80 hours of annotated speech). We use a classical PLP parameterization; feature vectors are composed of 12 coefficients and energy with the first and second derivative of these 13 coefficients. A cepstral normalization is performed in a 500ms sliding window. Two sets of speaker-independent acoustic models are used: a wide band model and a narrow band model, both gender-dependent.

The language models are classical trigrams estimated on about 330M of words from the French newspaper *Le Monde* 1987-2003 and the broadcast news training data (manual transcripts) provided during the ESTER campaign (960K words) with a vocabulary of 65K words. Language model includes 16.7M of bigrams and about 20M of trigrams. The system runs two passes. The first one provides intermediate transcripts which are used for Maximum Likelihood Linear Regression (MLLR) adaptation. The second transcription pass uses these MLLR models. The first pass takes about 3xRT and the second one about 5xRT on a standard desktop computer.

3.3. The LIA English broadcast news system - baseline

The English system is an adaptation of the previously presented French system.

The phone set used for the acoustic model is the same as the one used

in the Carnegie Mellon University Pronouncing Dictionary[1]. The acoustic model has the same structure as the French model: cross-word context-dependent acoustic models with 230k Gaussians and tying by decision trees using acoustic context related questions. The acoustic models are trained on the HUB4 English corpus [44] (about 70 hours of annotated speech). We use the same PLP parametrization as used for the French system.

The baseline LM for transcribing broadcast news is a 65k word classical 3-gram, estimated on 2.7G words from the Gigaword, North American News and HUB4 corpora, by using the modified Kneser-Ney smoothing technique. The weights of the corpora in the language model are chosen such as the perplexity is minimized on a small part of the development corpus of HUB4.

For transcribing surgery-related data from the AVISON corpus we used a combined 65k word LM, by interpolating general 3-grams learned on the HUB4 English corpus, with 3-grams estimated on all the reference transcriptions available in the AVISON training corpus. We rely, here as well, on the modified Kneser-Ney smoothing technique. The LM mix factors are chosen so that the perplexity of the resulting LM is minimized on a development corpus, which contains 5 hours of imperfect transcripts. The development set has been excluded from the training corpus. The acoustic models are the same as the HUB4 system.

We used the Carnegie Mellon University Pronouncing Dictionary for obtaining the phone representation of the lexicon words. The words that were not in this dictionary was automatically phonetized with Festival [45].

3.4. Confidence test

We use the following confidence test in our experiments [8]:

$$wer_f - u_{\frac{\alpha}{2}}\sqrt{\frac{wer_f(1-wer_f)}{k}} < wer_p < wer_f + u_{\frac{\alpha}{2}}\sqrt{\frac{wer_f(1-wer_f)}{k}} \quad (2)$$

Where k is the number of words in test, wer_f the WER on test. α defines a confidence interval: if $\alpha = 95\%$, then wer_p is defined with a confidence of $\pm 0.05\%$. $u_{\frac{\alpha}{2}}$ is defined by the *Student* value: $u_{0.425} = 1.96$.

We also propose the *Matched Pairs Sentence Segment Word Error Test* (mapsswe) score provided by the sc_stats NIST tool for WER results.

[1] http://www.speech.cs.cmu.edu/cgi-bin/cmudict

3.5. Anatomy of the Speeral decoder

The Speeral decoder [27] is derived from the A* search algorithm. A* is an algorithm dedicated to the search of the best path in a graph. It has been used in several speech recognition engines, generally for word-graph decoding. In Speeral, the search algorithm operates on phoneme lattice, which is estimated by using cross-word and context-dependent HMM.

The exploration of the lattice is supervised by an estimation function $F(h_n)$, which evaluates the probability of the hypothesis h_n crossing the node n:

$$F(h_n) = g(h_n) + p(h_n) \tag{3}$$

Where $p(h_n)$ is the probe that estimates the probability of the best hypothesis from the current node n to the ending node. In Speeral, the probe p combines the acoustic probability and a linguistic look-ahead score (LMLA - Language Model Look-Ahead) [29]. The acoustic term is determined via an acoustic decoding process, carried out as a Viterbi algorithm operating *backwards* (i.e., from the end of the signal, to its beginning), on the phone lattice. The LMLA used in Speeral enables the comparison of competing hypotheses before reaching a word boundary. The probability of a partial word corresponds to the best probability in the list of words sharing the same prefix:

$$P(W^*|h) = max_i P(W_i|h) \tag{4}$$

where W^* is the best possible continuation word and h the word history (partially present in $g(h_n)$). The LMLA approximation does not affect the results and speed-up the search process. This technique will be useful for Driven Decoding Algorithm to anticipate explored words. $g(h_n)$ is the probability of the current hypothesis that results from the partial exploration of the search graph (from the starting point to the current node n):

$$g(h_n) = \max P(W)^{\beta} \delta^{|W|} P(X|W) \tag{5}$$

Where $P(W)$ is the linguistic probability of the current word sequence, $P(X|W)$ is the acoustic probability according the word sequence W, β is the language model fudge factor (14 for French and 10 for English), δ is the linguistic penalty (-12 for French and -14 for English) and $|W|$ the number of words in the W sequence.

The best paths are then explored in a depth-first manner. This depth-first search refines the evaluation of the current hypothesis. Low-probability paths are stacked, thus allowing for a backtracking on these paths. In such situations, the search is desynchronized from the audio stream.

3.6. The Driven Decoding Algorithm

The Driven Decoding Algorithm aims to align and correct imperfect transcripts by using a speech recognition engine [25, 24, 23]. This algorithm improves the system performance by taking advantage of the availability of the approximate transcripts. This DDA integrates a DTW-based alignment within the A* algorithm. A first on-demand synchronization of the current explored hypothesis is performed with the prior transcript. Then, a transcript-to-hypothesis matching score is evaluated and used for fudging the n-gram probabilities.

3.6.1. ASR synchronization to the imperfect transcript

We assume that the presented synchronization is performed after a pre-segmentation stage. In the next experiments, we propose to use manual and automatic pre-segmentation stages.

The Speeral speech recognition system generates hypotheses as the phone-lattice is being explored. The best hypotheses at time t are extended according to the current hypothesis probability and the probe results.

In order to locate an anchoring point in the auxiliary transcript T of size m, each evaluated word from the current hypothesis H is aligned to T by using a Dynamic Time Warping (DTW [1]) algorithm.

In practice, a partial hypothesis H of size n is built by collecting the current word (i.e. the currently explored node, using LMLA) and its history from the path found during the search process. The sequence alignment is achieved by constructing a n-by-m matrix where the (i^{th}, j^{th}) element of the matrix contains the distance between the two words T_i and H_j. We use a basic distance function:

Considering a pre-segmentation stage, the auxiliary transcript size is close from the maximal hypothesis size.

$$\begin{array}{l} d(T_i, H_j) = 0 \text{ if } T_i = H_j \\ d(T_i, H_j) = 8 \text{ in the insertion cases} \\ d(T_i, H_j) = 6 \text{ in the deletion cases} \\ d(T_i, H_j) = 12 \text{ in the substitution cases} \end{array} \tag{6}$$

The deletion, insertion and substitution costs are computed via the estimated probability of each event in the approximated transcripts. The cumulative distance $\gamma(i, j)$ between H_j and T_i is computed as:

$$\gamma(i,j) = d(T_i, H_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \quad (7)$$

The alignment is performed at each newly encountered word in the phone lattice, during the search: a cache of the previous alignment is used in order to quickly increment or decrement the cumulative distance $\gamma(i, j)$.

Considering the low complexity of the word alignment with the cache, this on-demand synchronization process requires low computational resources. Moreover, the additional alignment cost may be balanced by the speed-up provided by well-transcribed sections (about 20% faster).

In Figure 1 we illustrate the dynamic synchronization of the search algorithm driven by the alignment on an imperfect transcript: for each explored node, the DTW algorithm allows one to align the imperfect transcript with the current generated hypothesis.

The synchronization robustness depends on the DTW alignment. If the partial hypothesis matches several repeated parts (this behavior is frequently observed at the segment beginning), the part with the best score is selected. However, after five words, the DTW path usually becomes stable.

The best hypothesis-to-reference matching provides a synchronization point that will be used to compute a matching score that is presented in the next subsection.

3.6.2. Weighting of the current hypothesis according to alignment

In order to use imperfect transcript information, the linguistic part of the $F()$ function is rescored according to a transcript-to-hypothesis matching score $\alpha(w)$. This mechanism drives the search by dynamically fudging $g(h_n)$, according to the alignment scores.

The matching score denoted α is based on the number of words in the short-term history that are correctly aligned with the transcript. α is greater when the trigram is aligned, and decreases with the misalignments of the history. The values of α were determined empirically by using a development corpus composed of a 15-minute show with a grid-search method:

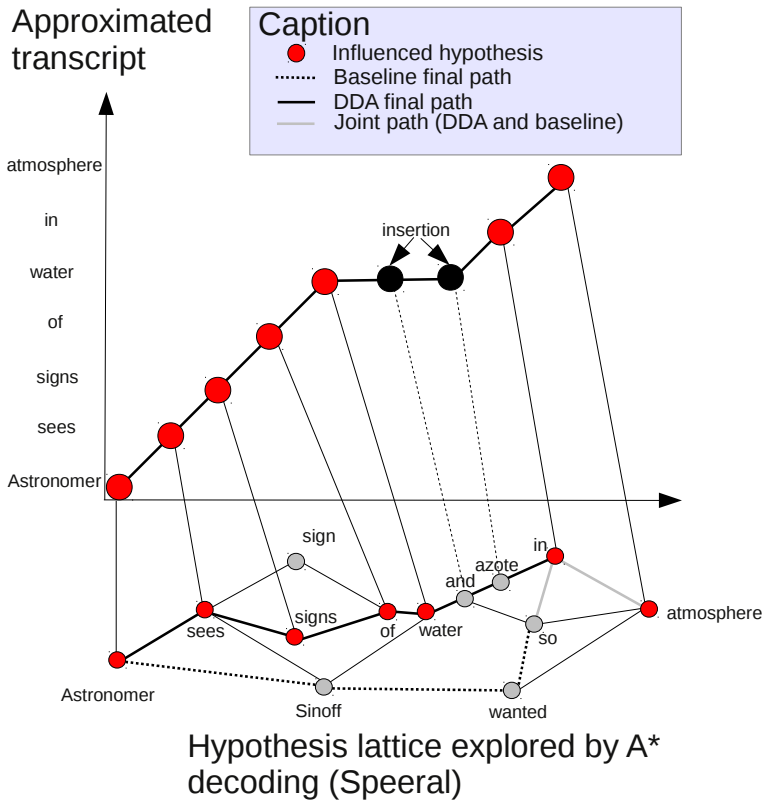

Figure 1: Synchronization of the search beams with the imperfect transcript by the DTW algorithm during an asynchronous decoding. Linguistic probabilities are fudged according to the quality of the alignment.

$$\alpha(W) = \begin{cases} 0.01 & \text{if } w_1, w_2, w_3 \text{ and more match the hypothesis} \\ 0.2 & \text{if } w_1 = t_1 \text{ and } w_2 = t_2 \text{ and } w_3 = t_3 \\ 0.4 & \text{if } w_1 = t_1 \text{ and } w_2 = t_2 \text{ and } w_3 \neq t_3 \\ 0.9 & \text{if } w_1 = t_1 \text{ and } w_2 \neq t_2 \text{ and } w_3 \neq t_3 \\ 0.99 & \text{if } w_1 \text{ is not found} \end{cases} \tag{8}$$

Where $(w_n, ..., w_1)$ are the words (left to right) of the currently explored hypothesis of the ASR system (w_1 is the word corresponding to the current node). $(t_n, ..., t_1)$ are the transcript words aligned to the current hypothesis. W is the whole word hypothesis.

These α values minimize the WER by using a simulated imperfect transcript (20% WER). Then, linguistic probabilities are modified by using the following fudging rule:

$$\tilde{P}(W^k) = P(W^k)^{\alpha(W^k)} \tag{9}$$

where k is the size of the currently explored word hypothesis W, $\tilde{P}(W^k)$ is the updated n-gram probability and $P(W^k)$ is the initial probability of the n-gram.

However, in practice, the α values have a limited impact (in a ± 0.5 range), while they decrease with the misalignments of the history. The α values presented in equation 8 are used in all our experiments.

With the DDA, equation 5 becomes:

$$\tilde{g}(h_n) = \max \prod_{k=1}^{k=|W|} P(W^k)^{(\beta+\alpha(W^k))} \delta^{|W|} P(X|W) \tag{10}$$

where $\alpha(W^k)$ is dynamically evaluated according to the imperfect transcript alignment with the currently explored hypothesis.

The set of experiments based on DDA in section 3.8 uses the mechanisms described above.

3.7. DDA with a synchronous Viterbi decoding framework

The proposed DDA can be easily transposed into a Viterbi decoding framework. The only requirement is to know the best history alignment of the currently explored word in the beam search, in order to fudge the linguistic score. This implementation requires another cost function inside the DTW algorithm and to compare multiple hypotheses at any time. The method proposed by [39] aligns closed captions in a similar way. The main difference in the Placeway approach is the rescoring trigram rule based on a conditional distribution $P(w_i|w_1...w_{i-1}; c_1...c_m)$ where w_i is the next word in a transcript whose closed-caption is $c_1...c_m$. This conditional distribution is inspired from [3], where the authors combine speech recognition and machine translation models. In our approach, the rescoring (or fudging) of linguistic rule is based on the history alignment, strongly biasing the ASR search. Moreover, in our framework, the best path is evaluated for each newly explored node: the ASR system is more robust to find matching segments.

3.8. Experiments

3.8.1. Experimental context

The experiments are carried out in the framework of the French ESTER evaluation campaign [13]. The ESTER corpus contains French radio broadcast news, including ad-hoc interviews, non-native speakers, on-the-fly translations, ... The results are reported on a test set of 4 hours from three broadcasters (France Inter 1 & 2, France Info and RFI), extracted from the official ESTER development set. The 3 : 1 balance between deletions and insertions was a choice during the ESTER evaluation campaign allowing the best WER. The language model perplexity is 132 for the whole test set. Only the first pass is assessed.

Baseline results are presented in Table 1:

Outline	Corr	Sub	Del	Ins	WER	S.Err	Conf. Int
FrInter 1	78.7	13.5	7.8	1.4	22.7	69.8	0.72
FrInter 2	79.9	12.3	7.2	1.6	21.1	65.7	0.70
FrInfo	78.6	13.2	8.2	2.9	24.3	67.8	0.76
RFI	76.2	15.5	8.3	3.4	27.2	57.6	0.83

Table 1: Details on the baseline system: Correct Rate (Corr), substitution rate (Sub), deletion rate (Del), Insertion rate (Ins), Word Error Rate (WER), Sentence Error rate (S.Err) and confidence Interval (Conf. Int)

3.8.2. Artificial transcripts

Here, the imperfect transcripts are generated by manually adding errors to the initial transcripts, while ensuring a correct journalistic form in order to respect the traditional style of a radio broadcast. In all cases, the meaning of sentences is preserved:

- We change some wording
- All disfluencies are removed
- Spontaneous speech is changed to formal speech
- Some words are replaced by synonyms

Outline	Corr	Sub	Del	Ins	WER	S.Err
FrInter 1	92.1	4.2	3.7	2.2	10.1	35.0
FrInter 2	91.4	3.6	5.0	1.6	10.2	65.7
FrInfo	85.9	8.1	6.0	6.2	20.3	54.9
RFI	92.3	4.4	3.3	2.3	10.0	43.9

Table 2: Details on errors added in the artificial transcripts: Correct rate (Corr), Substitution rate (Sub), Deletion rate (Del), Insertion rate (Ins), Word Error Rate (WER) and Sentence Error rate (S.Err)

Finally, 10% WER are introduced in three show transcripts, and 20% WER are introduced in the last show transcript. Table 2 presents the deletion, insertion and substitute rates in our artificial transcripts.

Here are some examples of imperfect transcripts (**ref**erence and **mod**ified transcripts):

- ref: vous écoutez france info il est midi
- mod: vous êtes à l'écoute de france info il est midi
- ref: le gouvernement n' évitera sans doute pas
- mod: le gouvernement n' évitera probablement pas
- ref: une entreprise américaine bechtel géant du btp
- mod: l' entreprise bechtel géant du bâtiment
- ref: un cadre de carrefour
- mod: un employé de carrefour
- ref: HEC et polytechnique
- mod: polytechnique et HEC
- ref: donc on tiendra (%HESITATION) ce qu' on a dit c' est à dire à l' heure actuelle... eh bien il n'y aura
- mod: donc on respectera ce qu'on a dit ... il n'y aura

However, we do not take normalization issues into account: some dates, numbers or named entities can have wrong forms.

3.8.3. Segmentation

In the experiments with artificial transcripts, we first run the decoder without the DDA in order to align the transcripts and the decoder output: alignment to the reference is performed by using the sclite NIST tool. Then, reference transcripts are segmented in advance with large margins (about 10 words before and after): for each audio segment, the corresponding imperfect transcript is known. These marginal words are added in order to increase robustness and to involve slight overlaps between reference segments. Moreover, timestamps are removed from the transcripts.

Segment duration is based on automatic speaker segmentation with a 30 seconds threshold (generally, segment length is from 5 to 30 seconds). The number of segments per hour is about 200. Table 3 reports details on automatic segmentation results.

Outline	#Segments	0-10 seconds	10-20 seconds	20-30 seconds
FrInter 1	220	49	30	101
FrInter 2	184	43	33	108
FrInfo	191	77	45	69
RFI	211	78	34	99

Table 3: Automatic segmentation for the development set: number of segments according to the duration bins

3.8.4. Language model interpolation

The linguistic variability can be reduced thanks to the contribution of an exact or imperfect transcript. Reducing the overall linguistic space helps improving the ASR performances. This can be achieved by estimating a language model on the transcript itself. However, such a language model would probably be too specific when the speaker deviates from the original transcript. Therefore, this model is interpolated with a generic language model, following the well-known linear combination scheme of language models (LMs).

Preliminary experiments are carried out in order to identify the potential benefits of the proposed methods. First, a language model is estimated with the accurate transcript. This model is combined with the generic 65K language model. These experiments measure the real effect of the proposed techniques on the decoder recognition performance: Out-Of-Vocabulary (OOV)

words are removed. In Table 4 we show the results of the interpolation of a language model trained on the exact transcript with the generic language model. For comparison, baseline decoding is performed by using the generic broadcast news language model. We obtain a WER of 22.7%.

Outline	WER
FrInter 1: LM-G only	22.7%
FrInter 1: LM-TrErr only	16.3%
FrInter 1: LM-TrEx only	5.2%
FrInter 1: LM-G 70% + LM-TrEx 30%	13.0%
FrInter 1: LM-G 50% + LM-TrEx 50%	11.5%
FrInter 1: LM-G 30% + LM-TrEx 70%	10.8%
FrInter 1: LM-G 70% + LM-TrErr 30%	16.2%
FrInter 1: LM-G 50% + LM-TrErr 50%	15.4%
FrInter 1: LM-G 30% + LM-TrErr 70%	15.2%

Table 4: Interpolation of the generic language model (LM-G) with a model trained on the exact transcript (LM-TrEx) and with the model trained on the imperfect transcript (LM-TrErr - 10% WER)

Then, a language model is generated by using the imperfect transcript (10% WER) as well. The experiments using this language model combined with the generic model are also presented in Table 4.

These experiments show that a decoding process based on a language model estimated on imperfect transcripts significantly improves WER. However, the WER remains higher than 10%, that corresponds to the initital WER of the imperfect transcript. The best results are obtained by combining the generic language model and the model estimated on the imperfect transcript.

Moreover, in the case of models trained on the exact transcript, the WER remains higher than 5%. Combined with a generic language model, the WER increases: transcript information is *lost*. Hence, we need to use transcript information more efficiently.

3.8.5. Experiments with model interpolation and Driven Decoding Algorithm

In this section we present experiments using the decoding strategy based on DDA. The experiments combining the interpolation of the language models with an alignment on the exact transcript are presented in Table 5.

Outline	WER
FrInter 1: LM-G + alTrEx	3.7%
FrInter 1: LM-TrEx + alTrEx	3.7%
FrInter 1: LM-G70%+LM-TrEx30%+alTrEx	3.7%
FrInter 1: LM-G50%+LM-TrEx50%+alTrEx	3.5%
FrInter 1: LM-G30%+LM-TrEx70%+alTrEx	3.7%

Table 5: Interpolation of the generic language model (LM-G) with the model trained on the exact transcript (LM-TrEx), and DDA with the exact transcript (alTrEx)

We obtain a minimum WER of 3.5%. This error rate can be considered as minimal for a concurrent hypothesis re-estimation method, without modifying the content of the hypothesis stack.

In Table 6 we show the experiments where the exact transcript is replaced with the imperfect transcript (10% WER).

Outline	WER
FrInter 1: LM-TrErr + alTrErr	9.9%
FrInter 1: LM-G + alTrErr	7.7%
FrInter 1: LM-G 70% + LM-TrEr 30% + alTrErr	7.2%
FrInter 1: LM-G 50% + LM-TrEr 50% + alTrErr	7.4%
FrInter 1: LM-G 30% + LM-TrEr 70% + alTrErr	8.6%

Table 6: Interpolation of the generic language model (LM-G) with the model trained on the imperfect transcript (LM-TrErr - 10% WER), and DDA with imperfect transcript (alTrErr - 10% WER)

Although this approach removes some of the limitations observed in the model combination, potential sources of error remain. In particular, heuristics are used in the decoder to reduce the search space and to speed up the decoding process. In ordinary conditions, the pruning introduces only a few errors; however when the acoustic context is of low quality, the best hypothesis can be excluded from the stack of available hypotheses. This occurs more frequently in real-time configurations of the system, when the pruning is stricter: a strategy highlighting the synchronized hypothesis rather than the others is not able to find the good hypothesis in the search space.

The best result is obtained by combining the generic language model

(with a 70% weight) with the model estimated on the imperfect transcript (with a 30% weight) and by carrying out a Driven Decoding with the latter. The DDA reduces the WER to 7.2%. This makes possible to add temporal information that is poorly taken into account by the language model. The use of DDA associated to the interpolation of the models results in a new gain.

In order to validate these results, we tested the best configuration on the four hours of test data. The results are reported in Table 7.

Shows	Baseline	Transcript	DDA
FrInter 1	22.7%	10.1%	7.4%
FrInter 2	21.1%	10.2%	7.7%
FrInfo	24.3%	20.3%	12.1%
RFI	27.2%	10.0%	7.3%

Table 7: WER obtained by the baseline system (*Baseline*), the original transcript (*Transcript*), the DDA

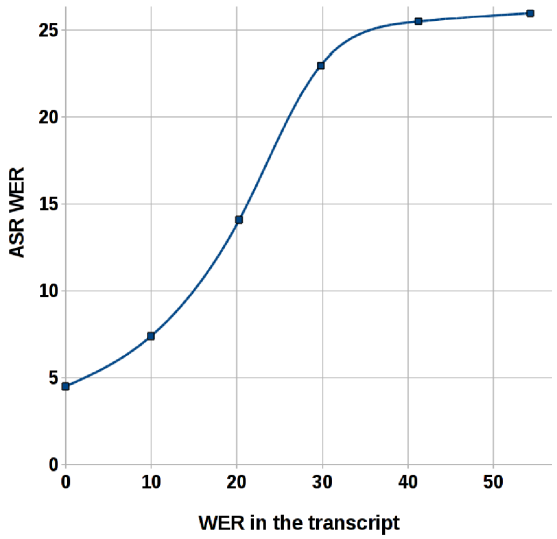

Figure 2: Impact of the WER variation on the RFI input transcript.

In order to compare the results on the same show, we introduced a WER from 0% to 55% (0, 10, 20, 30, 41 and 54) into the RFI transcript. The results

are plotted in Figure 2. We observe that the performance gain seems to be relatively independent of the quality of the initial transcript: the method is quite robust. These experiments show that imperfect information can be favorably used during the decoding process.

3.9. Discussion on the Driven Decoding Algorithm

A well-known advantage of the A* algorithm lies in the possibility to incorporate various information sources into the recognition process. However, it is an asynchronous algorithm and its application to alignment tasks can be difficult. We proposed an on-demand synchronization that allows the combination of asynchronous recognition and transcript-to-signal alignment. The system takes advantage of the approximate transcript as long as it allows for a gain and switches to free-recognition mode when the acoustic observations do not match the suggested transcript.

This method provides a significant gain in terms of WER even if the quality of the provided transcripts is low. The relative WER improvement obtained ranges between 28% and 40%. Moreover, we observe that the modified algorithm improves decoding speed slightly (by about 20%), in spite of the additional computational cost due to search synchronization. This gain in terms of execution time is due to the earlier exploration of the best paths on well-transcribed sections.

The work presented in this section relies on pre-segmented transcripts and the knowledge which imperfect transcript belongs to the segment. In real conditions, prompts or approximated transcripts do not contain accurate timestamps. Moreover, the data available is often incomplete and large. The lack of significant parts of transcript causes failures in the search for anchoring points. Therefore, the algorithm is not really relevant for the spotting of segments in large text corpora, without pre-segmentation. Sections 4 and 5 explore a dynamic extraction of segments in order to find parts matching the decoded speech. These matching parts will be called *transcript islands*. This dynamic extraction will eliminate the necessary first pass for pre-segmentation.

4. Fast hypothesis-to-transcript island matching: an integrated spotting task

The goal of our method is to take advantage of imperfect transcripts when they are available, while no timing information is available for the localiza-

tion of *transcript islands*. Moreover this method allows one to use the full prompts without pre-segmentation. Therefore, the main issue is to integrate, in the ASR system, an identification module that would be able to decide, at each node of the search graph, when the recognizer is crossing one of the available *transcript islands*. As the search graph is developed dynamically, this integration can be achieved by an on-the-fly spotting process.

The principle of the proposed method is close to approaches used in the field of information retrieval. Typically, search engines try to find the most relevant documents by comparing the query (i.e. the current explored hypothesis) to the indexed collection of stored documents. Most of the algorithms consist in building a set of ranked document lists. Here, we follow a similar scheme, while focusing on the efficiency of the algorithm.

The lexicon is represented by a lexical space where each dimension is associated to a word. All documents, including the hypothesis itself and the transcript, are represented in this lexical space by word-frequency vectors. The coefficients of these vectors represent the frequencies of the words in the document.

As the current hypothesis h is developed at a time t, a set of word clusters C_i is built and updated. These clusters result from the intersection of h and the transcript I. Start and end words of a cluster C_i delimit a *transcript island* I_i in the prompt. For each new word added to the hypothesis h, *transcript islands* are considered as candidates for guiding the search. This competition is arbitrated by a matching score S_i, which is computed as follows:

$$S_i(I_i) = \frac{|I_i|}{|h|} \sum_{w_k \in C_i}^{k} itf(w_k) \tag{11}$$

where $\mid I_i \mid$ and $\mid h \mid$ are the cardinalities of the island I_i and the current hypothesis h, respectively. C_i is a cluster resulting from the intersection between h and the approximated transcript as: $C_i = h \cap I_i$. w_k are the words of the current hypothesis and $itf(w_k)$ represents the inverse measure of the word frequency in the whole transcript:

$$itf(w) = \frac{1}{tf(w)} \tag{12}$$

where $tf(w)$ is the frequency of w in the document. This matching score represents a level of similarity of the hypothesis to the considered *transcript island*. This measure takes into account a semantic weight of the word, which depends on its relative frequency in the whole document.

If any matching score is higher than an *a priori* fixed threshold, the algorithm considers that it is on a *transcript island* and the search algorithm is driven by the corresponding word utterance: the *transcript island* becomes the imperfect transcript T as in section 3.6.1, then the search algorithm is driven by equation 9.

4.1. Algorithm development

This algorithm was developed for the Speeral ASR system. Despite of the deep-first search algorithm, in most case, only the last words of the current explored hypothesis are varying. This aspect allows us to control the computational cost of the proposed algorithm via the T_{freq} value. The threshold T_{freq} is based on word frequency and allows one to use only words appearing more than T_{freq}. In the proposed algorithm, N_{max} is the maximum number of desired clusters (100 is a good value). δ is the tolerance ($\pm 10\%$) of the maximum number of clusters. $tf(w)$ is the term frequency of the word w into the transcription. C_i is a cluster composed of the word positions $\{p_0, ..., p_m\}$. $\{C\}$ is the set of clusters. I_i is a *transcript island* delimited by the start (i.e. min position) and the end (i.e. max position) of a cluster C_i.

The algorithm is developed and detailed in the next page (Algorithm 1).

In practice, the set of cluster is not cleared for each new hypothesis. The process is incremental: a new hypothesis h differs often in the last words compared to the previous hypothesis h'. During the algorithm execution, all operations (creating a cluster, merging a cluster, adding a position in a cluster) are stacked. When a new hypothesis is explored, operations are unstacked to obtain the hypothesis $h'' = h \cap h'$. Then, algorithm is applied only on changing words.

However, if T_{freq} is varying, the set of cluster is then cleared and operations destacked.

```
Each word w is associated with its positions p in prompt;
Initialize N_max;
T_freq ← 0;
δ ← (10/100) N_max;
tf(w) is computed for each word w;
while The ASR system generates an hypothesis h = {w_1, ..., w_n} do
    {C} ← {};
    foreach w_j ∈ h satisfying condition tf(w_j) > T_freq do
        foreach position p_g of the word w_j into the prompt do
            foreach cluster C_i do
                if (min C_i{p_0, .., p_m}) − 2 ≤ p_g ≤ (max C_i{p_0, .., p_m}) + 5
                then
                    p_g is added to C_i;
                    if two clusters overlap then
                        they are merged;
                    end
                end
            end
            if p_g ∉ {C} then
                a new cluster is created with p_g;
            end
        end
    end
    if |{C}| > N_max + δ then
        T_freq ← T_freq + 1 ;
    end
    if |{C}| < N_max − δ then
        T_freq ← T_freq − 1;
    end
    S_i(I_i) is computed for each transcript island using equation 11;
    Transcript island with the best confidence score is selected.;
    The transcript island is used for DDA if |C_i|/|h| > 0.5 (i.e. half of
    words are similar);
end
```

Algorithm 1: Development of the text island spotter.

4.2. Island spotting experiments

4.2.1. Performance metric for spotting task

These experiments aim to evaluate the performance of the proposed method for building a high-quality speech corpus by using the closed-captions associated to a speech signal. On the spotting task, the system performance is evaluated in terms of precision/recall rates, on ESTER and RTBF ("Radio-Télévision Belge Francophone") corpora. F-measures are also reported. We first test the spotting performed on the ESTER corpus using gold and the degraded transcripts. Finally, we evaluate our method on the RTBF corpus based on real closed-captioning. Precision, recall and F-measure are defined as:

$$\text{precision} = \frac{\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}}{\{\text{retrieved documents}\}} \tag{13}$$

$$\text{recall} = \frac{\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}}{\{\text{relevant documents}\}} \tag{14}$$

$$\text{F-measure} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{15}$$

In the next ESTER experiments, the reference segments are viewed as documents. If the spotting process associates the correct reference segment (i.e. the relevant document) to the current decoded speech, the document is marked as well-retrieved. In the next experiments no DDA is used in order to focus on the spotting task

We simulate scattered data by randomly discarding 50% of the transcript segments. The removed transcripts have been chosen according to the reference speech segmentation. Finally, we obtain a reference with 500 to 800 annotated segments per hour (Table 8). The average duration of the remaining segments is about 6 seconds.

4.2.2. Spotting exact transcripts on the ESTER database

In the following experiments, DDA is not used. We evaluate the performance of our spotter. The experiments are performed on the four hours from the development set (i.e. France Inter 1 & 2, France Info and RFI). The spotting targets are *exact transcripts*. The results are reported in Table

8. We can see that, in these simulated conditions, spotting performance is good: more than 96.9% of the documents have been found, with a precision of about 94.4%. The results seem relatively independent of the performance of the ASR systems, which are varying, in this test, from 27.2% (the RFI show) to 22.7% (the France Inter show).

Radio station	Precision	Recall	F-measure	Doc. number	Conf. Int.
FrInter 1	90.9%	98.89%	94.8%	478	2.37
FrInter 2	94.3%	97.91%	96.1%	452	2.20
FrInfo	93.7%	92.9%	91.5%	468	2.87
RFI	98.9%	97.8%	98.4%	812	1.11
Mean	94.4%	96.9%	95.2%	2210	2.13

Table 8: Precision/recall, F-measure and confidence interval (for F-measure) of *transcript island* spotter on the exact transcript ESTER database. *Doc. number* is the total number of text segments based on manual annotation.

4.2.3. Spotting imperfect transcripts on ESTER database

As previously, 50% of transcript segments have been removed for spotting evaluation. The experiments are conducted on the four hours of the development set by using the imperfect transcripts (the same as those used in section 3.8.5).

These experiment aims at evaluating how errors in transcripts impact the spotting task performance. The results are reported in Table 9.

Radio	Precision	Recall	F-measure	Doc. number	Conf. Int.
FrInter 1	90.7%	96.9%	93.7%	478	2.54
FrInter 2	93.7%	96.7%	95.2%	452	2.37
FrInfo	93.4%	89.7%	91.5%	468	2.87
RFI	98.8%	97.8%	98.4%	812	1.11
Mean	94.2%	95.3%	94.7%	2210	2.22

Table 9: Precision/recall, F-measure and confidence interval (for F-measure) of *transcript island* spotter on the imperfect transcript ESTER database: 10% in FrInter 1 & 2, RFI and 20% in FrInfo. *Doc. number* is the total number of text segments based on manual annotation.

Comparing Table 8 and Table 9 we observe that the obtained precision and recall rates are very close to the ones obtained on perfect transcripts: the spotting task is robust to the increase of the WER.

4.2.4. Spotting real prompts on the RTBF corpus

The RTBF corpus has been collected in the framework of the AIDAR project [46]. It contains about 1000 hours of radio programs from the *Radio Télévision Belge*, in French and mostly recorded under clean conditions. Those programs mainly consist of news where topics and linguistic styles are rather close to those of the ESTER corpus. Among these 1,000 hours, the proportion of speech exceeds 300 hours (the remaining 700 hours are music, advertising or jingles) and prompts (provided in XML files) are available for about 60 hours of news. These prompts were used by the journalists. No further refinement on transcripts was done after recording, and no timestamps are available for precisely locating the speech segments corresponding to the provided transcripts. In order to evaluate our algorithm on this database, we have manually annotated the timestamps related to prompted paragraphs for 11 hours of the corpus, whereas all the 60 hours prompts are used. Audio without prompts is not used.

	Precision	Recall	F-measure	Par. number	Conf. Int.
RTBF shows	99.28 %	97.13 %	98.41 %	501	1.52

Table 10: Precision/recall, F-measure and confidence interval (for F-measure) of *transcript island* spotter. The experiment is performed on broadcast news shows from RTBF, by using the real journalist prompts. *Par. number* is the total number of paragraphs based on manual annotation.

In Table 10 we present experiments with our spotting technique on the 11 hours of annotated data. The results are better than those observed on the ESTER corpus: the average F-Measure is around 98%. 10 out of the 22 shows (of 30 minutes each) are fully timestamped at the paragraph level. These results are probably due to the fact that the prompt mapping is usually related to the global structure of the document, including speaker information, speech turns, non-speech segments, etc. Then, *transcript islands* match the natural segmentation of the document. Well-segmented transcripts are easier to spot. Moreover, paragraphs are significantly longer in the RTBF corpus: we have about 22 annotated paragraphs per show (30 minutes), for a total of 501; this limits the risk of missing an island that should be spotted.

The next section presents the combination between the *transcript island* algorithm and the Driven Decoding Algorithm.

5. Combining DDA and transcript island spotting

In previous sections, we presented two algorithms:

- The DDA method allows one to improve recognition rates by taking benefit of the available transcripts, even if they are not perfect.
- The *transcript island* spotting algorithm aims to find on-the-fly matching parts of large transcripts with the current hypothesis of the ASR system.

In this section, we evaluate the quality of the transcripts provided by the Speeral decoder guided by the prompts spotted on-the-fly into a large corpus: in the next experiments, prompt data are not pre-segmented.

The Figure 3 presents the spotter integration with the Driven Decoding Algorithm. The spotting system *transcript islands* extracts, from the large quantity of transcriptions and feeds them to the DDA.

5.1. Impact of DDA combined to transcript island spotting on the ESTER corpora

The ASR system is using together the spotting algorithm and the DDA. Two tests on four hours have been performed and compared to the baseline system (consisting of a classical Speeral run, without any helpful transcript): we first evaluate the WER by using segments of exact transcripts; then, the same experiment is performed on imperfect transcripts. We used previous imperfect transcripts of 10% to 20% WER, and 50% of the text segments have been removed for evaluating the spotting performance.

Table 11 reports the results obtained by the DDA search algorithm driven by the perfect transcript, and the DDA driven by imperfect transcripts: the DDA performance variance between Table 11 and Table 4 is due to the 50% of removed segments.

The results show that the driven recognizer takes advantage of the spotted segments: the ASR system is able to extract on-the-fly *transcript islands* and to use them via the DDA. As expected, the correct prompts remain more effective than imperfect ones; nevertheless, approximate transcripts yield a

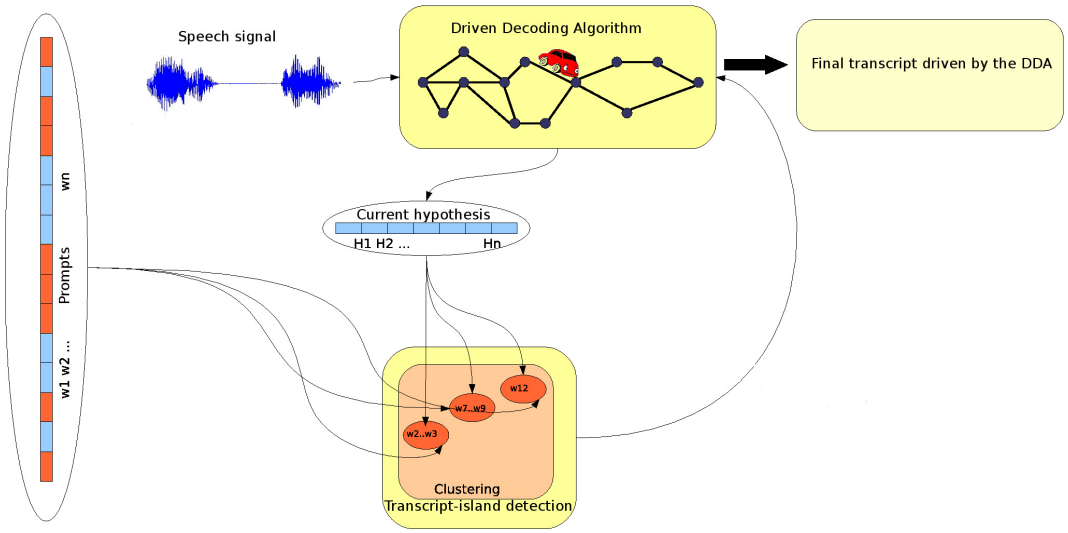

Figure 3: Scheme of the spotter integration with DDA. The *transcript island* detector computes an hypothesis-to-island matching score via a clustering algorithm. According to it, the spotter sets the decoder in driven-decoding mode. $w_1, .., w_n$ are the words provided by the approximated transcript, $H_1, .., H_n$ are the words of the current explored hypothesis in the $A*$ algorithm.

System	Baseline	DDA+IT	DDA+PT
FrInter 1	22.7 %	17.9%	17.1%
FrInter 2	21.1 %	16.6%	15.9%
FrInfo	23.4 %	21.7%	18.3%
RFI	27.2 %	23.0%	20.3 %
Mean	24.4 %	20.9 %	18.6 %

Table 11: WER of the systems involved in the experiments; DDA+IT consists in Driven-Decoding with imperfect transcripts; DDA+PT is the DDA search algorithm, driven by the correct word sequence. Moreover, 50% of the text segments have been removed.

WER gain of about 14% relative, while exact transcripts yield a WER gain close to 24% relative.

This set of experiments shows that the spotting algorithm combined with DDA is able to produce a better transcript. This technique takes advantage of all available information (i.e. audio stream and *transcript islands*): we obtain a better automatic transcription and a better alignment, thanks to

the decoder quality improvement. These two criteria allow one to increase the size of the correct ASR system transcripts in just one pass (without acoustic adaptation for a second pass). The search of text-segments is fast, on-the-fly and synchronized to the decoding process. In addition, experiments with more than one pass show that the system converges towards the best potential solution during the first pass.

5.2. Impact of DDA combined to transcript island spotting on the RTBF corpora, in real condition

We used 200 hours of speech signal with associated prompts. Prompts are grouped by month and are imperfect transcription of the spoken contents. This data allows us to measure the efficiency of our approach. A language model is estimated on all prompts (about 2.4 million words) and interpolated with a generic language model proportionately to the amount of data. This language model is used both for the baseline and DDA. Acoustic models are those used in previous experiments. The baseline is based on a slightly supervised decoding approach. Baseline results are presented in Table 12 for one pass decoding associated with an *a posteriori* alignment: we obtain about 30 hours of exact annotated speech. Then, we use the *transcript island* algorithm combined with the Driven Decoding algorithm, which allows us to assess the amount of the aligned data.

The *transcript island* algorithm allows one to use only one pass to align on-the-fly approximated transcripts with the ASR system.

System	Baseline	Driven Decoding
# Hours	200	200
# paragraphs	50370	50370
# decoded words	2 497 125	2 515 503
# spotted paragraphs	11158 (22%)	11487 (23%)
# aligned words in paragraphs	380 000 (15%)	615000 (25%)

Table 12: Number of matching words between the prompts and the ASR output (baseline system on the first column, island Driven Decoding on the second column).

The results show that with the *transcript island* Driven Decoding, 38% additional words are aligned to the prompts: we have 50 hours of exact annotated speech. We observe 25% of aligned prompts. The *data loss* can be explained by several aspects:

- An initial decoder with about 25% WER
- Parts of prompts are not pronounced
- Titles and abstracts are inside the prompt data
- The initial WER of prompts
- The prompt normalization is imperfect: date formats, OOV word pronunciation, named entities, ...

These results show a significant increase of the quantity of usable data. The DDA allows us to correct the ASR system on the fly: this increase of data quantity should result in a WER improvement. This experiment yields the expected results: a larger corpus with a potentially improved quality (imperfect transcripts are rarely in the ASR output).

The combination of the two proposed algorithms allows us to extract only aligned paragraphs and to improve the ASR output via the DDA (in Section 3.8, we show that the DDA always improves the baseline system and the imperfect transcripts). Consecutively, experiments on the broadcast news RTBF corpus show that the method produces a larger corpus than the baseline approach.

In order to evaluate the potential of the method for acoustic adaptation, we also performed experiments on a specialized-domain corpus, in the medical field (the AVISON corpus, described in the next section).

5.3. Experiments on the AVISON corpus: acoustic adaptation

The AVISON corpus contains around 48 hours of commented English surgical intervention films. The spoken material in this corpus contains speech in several registers: read speech documenting surgical issues, spontaneous descriptions of surgical interventions, or spoken dialogues between surgeons and students. We have only imperfect transcripts for these audio documents. The AVISON corpus also contains a collection of textual documents related to surgery (scientific articles, surgery proceedings, protocol descriptions, etc.), which can be used for language model training purposes. We build a test corpus by manually transcribing four hours of this corpus.

On these four hours, comparing the initial imperfect transcript WER is 11.7% (sub = 1.56 % / del = 3.14 % / ins = 6.98 %). The language model perplexity is 127 on the test corpus. We present some examples of errors (**tra**nscription and **ref**erence):

- tra: the thyroid space here the carotid artery is here ********** it is
- ref: the ****** ***** **** *** carotid artery is here carotide artery is
- tra: the common **** *** side of
- ref: the common bile duct side of
- tra: cholecystitis **** *** when the plane between
- ref: cholecystitis when the when the plane between
- tra: are now slowly reaching ***
- ref: are **** slowly reaching now
- tra: cystic duct very rapidly here you see the anatomy
- ref: cystic duct **** ****** here you see *** anatomy

For building a transcription system for this kind of data, we used the transcription system, Speeral, that was described in the previous Section 3.2.

The system used for transcribing this domain-specific corpus is described in Section 3.3. A baseline WER of 41.6%, with unsupervised speaker adaptation, was obtained with this system on the AVISON test corpus: a first pass is used to adapt acoustic models with the MLLR technique (no confidence threshold is used for the Baseline adaptation). Moreover, 5% of the test words are OOV.

In order to improve the baseline acoustic model, we used techniques for automatically obtaining aligned speech and references from the spoken documents, where imperfect transcripts are available. These aligned data are then used for adapting the acoustic models to the speech conditions of the database. We tried two approaches, the first one consists in decoding the speech documents, and then aligning the result of the decoding with the imperfect transcripts. The alignment algorithm used is DTW. The second one consists in driving the decoding of the speech documents with the imperfect transcripts and thus producing aligned data. In the two cases, the LM used for decoding (first and second pass) is interpolated with the imperfect transcripts themselves. The lexicon used contains all the words that are present in the imperfect transcripts.

The amount of aligned data obtained with the two approaches is reported in Table 13. The Table also contains the ASR system accuracy, in terms of WER, on the AVISON test corpus. These performance figures have been obtained by using the acoustic models adapted with the aligned data.

System	Aligned data	WER after Adapt.	Conf. Int.
First Pass	nothing	45.6%	0.50
MLLR adaptation	48h	41.6 %	0.49
Baseline [22]	9h48	32.0 %	0.46
DDA	11h38	31.2 %	0.46

Table 13: Quantity of aligned data with slightly supervised adaptation (Baseline) or with the DDA, and impact of the aligned data on acoustic model adaptation

The first point to note is that the DDA increases the quantity of aligned data by 18%, compared to a classic *a posteriori* alignment: from the around 10 hours obtained without the DDA, we obtain around 12 hours by using the DDA. This increase of the amount of training data yields a gain of 0.8% WER absolute, which represents a significant relative gain of 2.5%. The *mapsswe* test finds a significant difference at the level of $p = 0.01$.

Table 14 presents more details about encountered errors in the baseline, DDA-based adaptation and slightly supervised adaptation.

Outline	Corr	Sub	Del	Ins	WER
Baseline P1	65.4	28.7	5.8	11.0	45.6
Baseline P2	71.1	24.4	4.4	12.8	41.6
Slightly sup. P1	76.9	19.4	3.6	10.1	33.1
Slightly sup. P2	78.8	17.9	3.1	10.8	32.0
DDA P1	77.0	19.4	3.6	9.4	32.4
DDA P2	79.0	17.6	3.2	10.3	31.2

Table 14: Precision on encountered errors: Correct Rate (Corr), substitution rate (Sub), deletion rate (Del), insertion rate (Ins) and word error rate (WER). P1 and P2 are respectively the first and second pass of the ASR system

5.4. Acoustics models trained from scratch on the AVISON corpus

In this section, we present experiments where the ASR system is trained from scratch by using the generated corpora. As previous, our baseline is

based on the lightly supervised approach. We have decoded the 48 hours of AVISON corpus using lightly supervised training approach (Baseline) and the DDA approach (DDA-full). Then, we have trained acoustic models on these two sets. The results obtained on the test corpus with these adapted models are reported in Table 15.

Outline	Corr	Sub	Del	Ins	WER	Conf. Int.
Baseline	71.7	24.5	3.9	11.7	40.0	0.49
DDA-full	73.2	23.0	3.7	11.6	38.4	0.49
DDA-random	71.5	24.6	3.9	11.9	40.5	0.49

Table 15: Acoustic models trained from scratch: Correct Rate (Corr), substitution rate (Sub), deletion rate (Del), insertion rate (Ins), word error rate (WER) and confidence interval (Conf. Int.). Baseline is a lightly supervised training approach, DDA-full use the same data sources, DDA-random use randomly selected segments in order to obtain the same amount of data that the Baseline

The results show a 4% relative WER improvement compared to the baseline. This aspect highlights that the DDA training database is significantly better (the *mapsswe* test finds a significant difference at the level of $p = 0.001$.) for training compared to the slightly supervised training. However, these experiments do not take information about the improvement causes: the larger amount of data available and/or the better quality of the transcripts.

In a second way, we randomly remove segments in the DDA generated data in order to obtain the same amount of training data that the slightly supervised approach. Then, we have trained the acoustic models. The results are presented in the last line of Table 15. The results are similar to the Baseline: lightly supervised training approach and DDA approach generates data of equivalent quality.

6. Conclusion

We have presented a method that aims to improve imperfect transcripts by using ASR technology. The proposal consists mainly in a Driven Decoding Algorithm that combines asynchronous recognition and transcript-to-signal alignment.

The system takes advantage of the approximate transcripts as long as they match the speech contents, and switches to free-recognition mode when the

acoustic observations do not match the suggested transcript. Performance is evaluated step-by-step, by comparing systematically the results obtained to the potential gains estimated by oracle measures.

The results demonstrated the efficiency of this technique, although the quality of the provided transcripts is low: the relative WER improvement is between 28% and 40%, compared to the initial prior transcript. Moreover, we observe that the modified algorithm improves slightly the decoding speed, in spite of the additional computational cost due to the search synchronization.

Partial transcriptions, corresponding to the situation where prompted and unprompted speech segments alternate, are handled by a spotting mechanism integrated in the DDA. This algorithm is inspired from information retrieval techniques. Its role consists in dynamically detecting *transcript islands* when the recognizer encounters them. Our experiments have shown that the proposed technique yields very good results by using real prompts provided with the RTBF database. Moreover, this method seems to be quite robust to imperfect transcripts.

Finally, we have proposed a method that allows one to synchronize an imperfect transcript on the fly and to drive the ASR output with them, allowing it to determine the missing timestamps and to generate transcripts closer to the audio stream than slightly supervised algorithm [22]: with the same source of corpora, DDA approach produces a larger amount of data.

Prospects of improvement are related to the use of the generated corpus as source and target of the system adaptation process: by performing a system adaptation to the local context of a speech segment, one can expect a significant improvement of ASR system accuracy on the targeted segment. The quality of the transcript may be improved by such a recursive self-training of the system, that may be viewed as a local version of the slightly supervised algorithm [22].

More generally, Driven Decoding offers a generic scheme for the integration of a text stream into the search algorithm. In previous papers, we studied how this paradigm may be used for ASR system combination [26] or for query-driven term spotting [40]. We now plan to improve the usage of highly uncertain information sources by integrating confidence scores and temporal dependencies in the DDA-based framework.

[1] Berndt, D., Clifford, J., 1994. Using dynamic time warping to find patterns in time series. In: Workshop on Knowledge Discovery in Databases (KDD'94). pp. 359–370.

[2] Bonastre, J.-F., Wils, F., Meignier, S., 2005. Alize, a free toolkit for speaker recognition. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05). Vol. 1. pp. 737–740.

[3] Brown, P., Chen, S., Pietra, S. D., Pietra, V. D., Kehler, S., Mercer, R., 1994. Automatic speech recognition in machine aided translation. Journal: Computer Speech and Language 8, 177–187.

[4] Cardinal, P., Boulianne, G., Comeau, M., 2005. Segmentation of recordings based on partial transcriptions. In: Proc. Interspeech'05. pp. 3345–3348.

[5] Chan, H. Y., Woodland, P., 2004. Improving broadcast news transcription by lightly supervised discriminative training. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04). Vol. 1. pp. 737–40.

[6] Chen, L., Gauvain, J.-L., Lamel, L., Adda, G., 2004. Dynamic language modeling for broadcast news. In: Proc. International Conference on Spoken Language Processing (ICSLP'04). pp. 1281–1284.

[7] Chen, L., Lamel, L., Gauvain, J.-L., 2004. Lightly supervised acoustic model training using consensus networks. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04). Vol. 1. pp. 189–92.

[8] Chollet, G., 1995. Evaluation of asr systems, algorithms and databases. Speech recognition and coding: New advances and trends, 32–40.

[9] Clarkson, P., Robinson, A., 1997. Language model adaptation using mixtures and an exponentially decaying cache. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'97). Vol. 2. pp. 799–802.

[10] Cormen, T., Leiserson, C., Rivest, R., Stein, C., 2001. Introduction to algorithms. MIT Press.

[11] Finke, M., Waibel, A., 1997. Flexible transcription alignment. In: Proc. IEEE Workshop Automatic Speech Recognition and Understanding (ASRU'97). pp. 34–40.

[12] Fiscus, J. G., Ajot, J., Garofolo, J. S., 2008. The rich transcription 2007 meeting recognition evaluation. Multimodal Technologies for Perception of Humans: International Evaluation Workshops CLEAR'07 and RT'07, 373–389.

[13] Galliano, S., Geoffrois, E., Mostefa, D., Choukri, K., Bonastre, J.-F., Gravier, G., 2005. The ester phase 2 based evaluation campaign for the rich transcription of french broadcast news. In: Proc. of the European Conference on Speech Communication and Technology (ICSLP'05). pp. 1149–1152.

[14] Haubold, A., Kender, J. R., 2007. Alignment of speech to highly imperfect text transcriptions. In: Kender, J. R. (Ed.), Proc. IEEE International Conference on Multimedia and Expo (ICME'07). pp. 224–227.

[15] Hull, D. A., 1996. Stemming algorithms: A case study for detailed evaluation. Journal of the American Society of Information Science 47, 70–84.

[16] Ibrahimov, O., Sethi, I. K., Dimitrova, N., 2002. Clustering of imperfect transcripts using a novel similarity measure. Journal: Information Retrieval Techniques for Speech Applications 1, 23–34.

[17] Iyer, R., Ostendorf, M., Jan. 1999. Modeling long distance dependence in language: topic mixtures versus dynamic cache models. Journal: IEEE Transactions on Speech and Audio Processing 7, 30–39.

[18] Jelinek, F., 1990. Self-organized language modeling for speech recognition. Journal: Language Processing for Speech Recognition, 450–506.

[19] Kemp, T., Waibel, A., 1999. Unsupervised training of a speech recognizer: Recent experiments. In: Eurospeech'99. pp. 2725–2728.

[20] Keogh, E., Pazzani, M., 2001. Derivative dynamic time warping. In: International Conference on Data Mining (SDM'01).

[21] Kuhn, R., De Mori, R., 1990. A cache-based natural language model for speech recognition. Journal: IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 570–583.

[22] Lamel, L., Gauvain, J.-L., Adda, G., 2002. Lightly supervised and unsupervised acoustic models training. Journal: Computer Speech and Language 16, 115–229.

[23] Lecouteux, B., Linarès, G., 2008. Using prompts to produce quality corpus for training automatic speech recognition systems. In: Proc. 14th IEEE Mediterranean Electrotechnical Conference (MELECON'08). pp. 841–846.

[24] Lecouteux, B., Linarès, G., Beaugendre, F., Nocéra, P., 2007. Text island spotting in large speech databases. In: Interspeech'07. pp. 1318–1321.

[25] Lecouteux, B., Linarès, G., Bonastre, J., Nocéra, P., 2006. Imperfect transcript driven speech recognition. In: InterSpeech'06. pp. 1626–1629.

[26] Lecouteux, B., Linarès, G., Estève, Y., Mauclair, J., 2007. System combination by driven decoding. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'07). Vol. 4. pp. 341–344.

[27] Linarès, G., Nocéra, P., Massonié, D., Matrouf, D., 2007. The lia speech recognition system: from 10xrt to 1xrt. In: Proc. of the 10th international conference on Text, Speech and Dialogue (TSD'07). pp. 302–308.

[28] Martins, C., Teixeira, A., Neto, J., 2007. Dynamic language modeling for a daily broadcast news transcription system. In: Proc. Automatic Speech Recognition & Understanding IEEE Workshop (ASRU'07). pp. 165–170.

[29] Massonié, D., Nocéra, P., Linarès, G., 2005. Scalable language model look-ahead for lvcsr. In: Proc. of InterSpeech'05. pp. 569–572.

[30] Mohri, M., 2002. Edit-distance of weighted automata. In: Conference on Implementation and Application of Automata (CIAA'02). pp. 1–23.

[31] Moreno, P. J., Joerg, C., Thong, J.-M. V., Glickman, O., 1998. A recursive algorithm for the forced alignment of very long audio segments. In: International Conference on Spoken Language Processing (ICSLP'98).

[32] Ney, H., Essen, U., Kneser, R., 1994. On structuring probabilistic dependencies in stochastic language modeling. Journal: Computer Speech and Language 8, 1–38.

[33] Nguyen, L., Xiang, B., 2004. Light supervision in acoustic model training. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '04). Vol. 1. pp. 185–188.

[34] Paulik, M., Fügen, C., Stüker, S., Schultz, T., Schaaf, T., Waibel, A., 2005. Document driven machine translation enhanced asr. In: Proc. Interspeech'05. pp. 2261–2264.

[35] Paulik, M., Waibel, A., 2008. Lightly supervised acoustic model training on epps recordings. In: Proc. Interspeech'08. pp. 224–227.

[36] Petrik, S., Kubin, G., 2007. Reconstructing medical dictations from automatically recognized and non-literal transcripts with phonetic similarity matching. In: Proc. IEEE International Confeference on Acoustics, Speech and Signal Processing (ICASSP'07). Vol. 4. pp. 1125–1128.

[37] Petrik, S., Pernkopf, F., 2008. Automatic phonetics-driven reconstruction of medical dictations on multiple levels of segmentation. In: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08). pp. 4317–4320.

[38] Placeway, P., Chen, S., Eskenazi, M., Jain, U., Parikh, V., Raj, B., Ravishankar, M., Rosenfeld, R., Seymore, K., Siegler, M., Stern, R., Thayer, E., 1997. The 1996 hub-4 sphinx-3 system. In: Proc. of the 1997 ARPA Speech Recognition Workshop. pp. 85–89.

[39] Placeway, P., Lafferty, J., 1996. Cheating with imperfect transcripts. In: Proc. International Conference on Spoken Language (ICSLP'96). Vol. 4. pp. 2115–2118.

[40] Rouvier, M., Linarès, G., Lecouteux, B., 2008. On-the-fly term spotting by phonetic filtering and request-driven decoding. In: Proc. IEEE Spoken Language Technology Workshop (SLT'08). pp. 305–308.

[41] Salton, G., 1988. Automatic Text Processing. Addison-Wesley Longman Publishing Company.

[42] Salton, G., Buckley, C., 1988. Term-weighting approaches in automatic text retrieval. Journal: Information Processing & Management 24, 513–523.

[43] Smith, T. F., Waterman, M. S., 1981. Identification of common molecular subsequences. Journal: Molecular Biology 147, 195–197.

[44] Stern, R., 1997. Specifications of the 1996 hub-4 broadcast news evaluation. In: Proc. of the DARPA Speech Recognition Workshop.

[45] Taylor, P., Black, A., Caley, R., 1998. The architecture of the festival speech synthesis system. In: Proc. of the third ESCA Workshop in Speech Synthesis. pp. 147–151.

[46] Tshibasu-Kabeya, B., Bontempi, G., Beaugendre, F., Marechal, G., 2006. Aidar : Une architecture pour l'indexation de documents audio numériques. In: Proc. Veille Stratégique Scientifique & Technologique (VSST'06).

[47] Wagner, R., Fisher, M., 1974. The string-to-string correction problem. The journal of the ACM 1, 168–173.

[48] Wessel, F., Ney, H., 2005. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. Journal: IEEE Transactions on Speech and Audio Processing 13, 23–31.

[49] Witbrock, M. J., Hauptmann, A. G., 1997. Using words and phonetic strings for efficient information retrieval from imperfectly transcribed spoken documents. In: Proc. of the second ACM international conference on Digital libraries (DL'97). pp. 30–35.

[50] Witbrock, M. J., Hauptmann, A. G., 1998. Improving acoustic models by watching television. Tech. rep., CMU-CS-98-110, Carnegie Mellon University.

[51] Woodland, P., Povey, D., 2002. Large scale discriminative training of HMM for speech recognition. Journal: Computer Speech and Language 16, 25–47.

[52] Yu, K., Gales, M., Wang, L., Woodland, P. C., 2010. Unsupervised training and directed manual transcription for lvcsr. Speech Communication 52, 652–663.