INTERSPEECH 2009 BRIGHTON

JTrans: an open-source software for semi-automatic text-to-speech alignment

C. Cerisara, O. Mella, and D. Fohr

LORIA INRIA - UMR 7503 - France

Abstract

Aligning speech corpora with text transcriptions is an important requirement of many speech processing, data mining applications and linguistic researches. Despite recent progress in the field of speech recognition, many linguists still manually align spontaneous and noisy speech recordings to guarantee a good alignment quality. This work proposes an open-source java software with an easy-to-use GUI that integrates dedicated semi-automatic speech alignment algorithms that can be dynamically controlled and guided by the user. The objective of this software is to facilitate and speed up the process of creating and aligning speech corpora.

Index Terms: JTrans, alignment, Transcriber, speech recognition

10.21437/Interspeech.2009-531

1. Introduction and motivations

JTrans is an open-source software for semi-automatic text-to-speech alignment[1]. In addition to the basic manual editing functionalities that are common in similar tools, JTrans further proposes dedicated semi-automatic speech alignment algorithms that can be dynamically controlled and guided by the user.

The objective of this paper is twofold:

- let the speech processing community know about JTrans and about its unique set of features, summarized in section 3.2, which makes it complementary to other softwares like Transcriber [1] or WinPitch [2].
- describe an original approach, alignment confidence measure adaptation, which is integrated in JTrans, to exploit user feedback in order to continuously improve JTrans efficiency as it is used.

Computer-aided speech transcription is a research area that has received a lot of attention recently. Our objective is rather to strengthen the computer-assisted speech alignment domain, which addresses more specific applications. This work actually originates from a request from linguist researchers, who needed efficient tools to precisely align long audio recordings with manual transcriptions that were already available. The main application requirements are word-level alignment for anonymization[2], support for low quality recordings, alignment of spontaneous speech, user control of the alignment process and easy ready-to-use GUI. Besides these specific needs, the proposed solutions might relatively easily be extended and applied to other application domains, such as speech transcription from scratch.

[1]Please visit `http://www.loria.fr/~cerisara/jtrans/` to download JTrans with additional details, screenshots, workflows, ...

[2]e.g., beeping audio segments that match proper nouns.

2. Related works

2.1. Related works about automatic alignment algorithms

Automatic alignment of a text and speech sentence is usually realized with a particular implementation of the Viterbi algorithm that is known as "forced alignment" algorithm [3]. However, this algorithm requires to store in memory the whole decoding trellis and is thus not suitable to handle very long audio files. Several solutions have been proposed to address this issue: most of them are now derived from the "anchor-based" approach proposed in [4]. This method exploits several speech recognition passes to find out reliable "anchors" between the text and the audio file. Typically, an anchor is set when a sequence composed of several words of the recognized transcription matches the same sequence in the input text: these words are likely to be exact, and the corresponding alignment defines an anchor. The speech recognition passes are realized with a vocabulary and grammar restricted to the words that appear in the text to increase robustness to noise. Once the distance between two anchors is small enough, a final pass of forced alignment between these two anchors can be realized. This approach is also used in [5] as a building block of a more general system for audio indexing and semantic processing.

The issue of aligning highly imperfect text to speech has been specifically addressed in [6], where recognition is first performed at the phoneme level only with monophthongs and fricatives, which appear to be more robust to noise than other phonemes.

The authors of [7] proposed a similar approach to [4], but without recursive application of speech recognition, and with an additional final pass that corrects the input text by replacing the insertions and substitutions between the text and the automatic transcription with words proposed by a large-vocabulary speech recognizer. This approach provides very good results, but on a corpus composed of recorded lectures with an initial speech recognition WER below 25 %.

Our corpus is composed of very bad quality recordings (spontaneous conversations in a train, in a cafe, ...), but with a reasonably good initial manual transcription. Furthermore, in our application, linguist researchers require an accurate control over the whole process, and fully automatic processing do not fit these requirements. We focus in our solution on the *interactions* between the user and semi-automatic tools, and how to exploit these interactions to improve the performances of the alignment tools.

2.2. Available alignment softwares

Two classes of software exist to perform text-to-speech alignment:

- toolboxes, such as HTK, which implements some of the aforementioned alignment algorithms. But these toolboxes do not fulfill our requirements, and are designed

for speech recognition researchers. They are hardly suitable for other users, and in particular linguistic researchers.

- ready-to-use softwares, and in particular *Transcriber*, *WinPitch*, and *Praat*.

In the latter class, the only software that proposes an automatic text-to-speech alignment is *Praat* with the plugin *EasyAlign* [8]. However, this solution is limited to very short speech segments.

WinPitch proposes an original mode for manual alignment that slows down the playback speed, in order to let the user more time to define anchors accurately.

Transcriber is a widely-used software to manually align text to speech. But it is not designed to align quickly with a great precision, at the word-level for example.

3. JTrans features

3.1. Basic features

JTrans supports most of the common functionalities included in signal and text editing applications, such as navigation in long audio files, text edition and audio segments definition, modification and alignment with text.

Meta-data (speaker, comments, noise, punctuation, simultaneous speech, ...) can be coded within the text itself: a customizable text parser based on regular expressions is used to preprocess and highlight these meta-data before automatic alignment.

In addition, JTrans proposes several speech processing algorithms, including a classical forced alignment Viterbi and French phonetizer, but also an incremental block-Viterbi and an adaptable alignment confidence measure.

Interaction with the user is a key feature of JTrans, and is designed to be efficient in both directions: thanks to automatic processing, the user makes less efforts and can progress faster; thanks to user feedback, confidence models continuously improve. JTrans proposes two GUIs: the first one, dedicated to fast and coarse alignment, is based on the "text metaphor" (i.e., without signal visualization), and only shows a text editing pane with playing and synchronization facilities. The second one, dedicated to manual fine-tuning, is based on the "timeline" metaphor and further displays the signal on top of the text.

3.2. Originality of the work

This work proposes two main original algorithms: (i) a unique "guided semi-automatic incremental alignment" approach described in section 4, and (ii) an original alignment confidence measure and its incremental adaptation described in section 5.

JTrans further proposes the following unique features:

- fastest coarse-grain semi-automatic alignment available;
- first semi-automatic software that exploits the user feedback for continuous improvement in the domain of text-to-speech alignment;
- smooth integration of automatic speech processing algorithms within a dedicated GUI for a total control of the user over these algorithms in real-time;
- cross-platform open-source software written in pure Java, with support for extensible sound formats (MP3, ...) thanks to the Java Service Provider Interface;
- support for unlimited audio file length with a central circular buffer;
- support for unlimited lexicon thanks to (i) a phonetized version of the Morphalou lexicon[3], and (ii) a rule-based phonetizer for out-of-vocabulary words;
- support for a subset of TRS/STM Transcriber formats for input/output, with the objective to support in the near future a richer standard format such as TEI[4].
- open plugins-centered architecture to integrate advanced speech recognition algorithms. Although the present version of JTrans only supports French text-to-speech alignment, it is very easy to extend to other applications, for example small-vocabulary speech recognition.

4. Guided incremental alignment

The whole alignment process can be done in three steps:

- fast semi-automatic alignment, with the help of automatic algorithms to reach a faster than real-time processing. The output of this step is a coarse-grain alignment at the sentence level;
- batch automatic processing that enhances the result of the previous step into an accurate word-level alignment. The output of this step is a fine-grain alignment at the word or phoneme level;
- final pass of manual checking and fine-tuning.

The first step is based on a fast incremental block-Viterbi alignment algorithm that is controlled by the user, who can easily switch between computer-aided and fully manual modes at any time. This step exploits a dedicated GUI based on the "text metaphor", which is best suited for fast and coarse processing [9]. In order to facilitate text processing by the user, a strict chronological left-to-right approach has been adopted at this stage.

Let us assume that text and speech are aligned up to time t.

- The block-Viterbi algorithm incrementally processes the speech signal until 20 words are aligned, using a 5s-length sliding window with a 2.5s shift.
- The last three seconds of the speech signal are then played back with synchronized text highlighting of the uttered words; the user can thus immediately check whether the alignment is correct or not;
- If the user agrees with the alignment, he simply presses "Enter" and the algorithm stores an anchor at the end of the block and proceeds to the next block. Otherwise, the user presses "ESC" to enter fully-manual mode. He can later on press "F5" to launch the block-Viterbi algorithm again when desired.

In fully-manual mode, speech is simply played and the user clicks on the words that have just been uttered, to put anchors. This ergonomic solution has been chosen because it minimizes the number of user manipulations and can be performed in real-time, as long as an accurate alignment is not required; this is the case with JTrans that postpones fine-tuning of this coarse alignment to the second and third steps.

The second step is realized in the background fully automatically by applying classical forced alignment algorithms between two successive anchors.

The third (optional) step is done manually by adjusting the segments limits, using another GUI based on the "timeline"

[3] `http://www.cnrtl.fr/lexiques/morphalou/`
[4] `http://www.tei-c.org/index.xml`

metaphor that shows both the audio signal and the corresponding text.

In the best case, with correct automatic alignment, the user only has to listen to a fraction of the actual audio corpus. A faster than real-time processing is then achievable. In the worst case, the automatic alignment performs poorly, and the user will then fall back to the fully-manual mode, which can be realized in real-time.

5. Confidence measure adaptation

5.1. Motivation

Compared to most other traditional alignment algorithms, JTrans is designed to work in interaction with the user. This feature opens up the possibility to develop and integrate in JTrans many solutions to continuously improve the automatic algorithms by exploiting the user feedback.

We have explored such a possibility by designing a confidence measure that continuously adapts to the signal thanks to the information provided by the user. This information can take two forms: either a "confirmation anchor" when the user presses ENTER to confirm that the automatic alignment is correct, or a "correction anchor" when he clicks on a given word with the mouse.

To the best of our knowledge, this is the first time a confidence measure is developed for the text to speech alignment task, and also that the user feedback is used to adapt this confidence measure.

The objective of this confidence measure is to estimate, for any new speech segment, whether the automatic aligner will perform correctly or not on this segment. When the automatic aligner is likely to get wrong, then the system will skip this suspicious speech segment and automatically align the next segment.

The rationale behind this algorithm is to tackle a weakness of the block-Viterbi method that fails to process the signal further when a very low-quality speech segment occurs. Such a situation occurs in our corpus for instance when someone is laughing loudly while another person is speaking, or when someone is talking too far from the microphone, which results in a hardly audible mumbling speech. By skipping this segment, the aligner may resynchronize the next higher quality audio segment with the text in order to find the next anchor more easily. The skipped segments will be realigned in the second stage, by forced alignment between the previous and next anchors. This solution is similar to the "usable speech" approach, which removes the speech segments that might induce errors in a speaker recognition task, or to the "missing data recognition" paradigm that removes unreliable features for speech recognition.

5.2. Confidence measure definition

The confidence measure takes as inputs a speech segment and outputs a score that represents how likely is the automatic aligner to be correct. The score corresponds to a log-likelihood ratio between two models, respectively for correctly and incorrectly aligned segments.

The user feedback is exploited by updating the parameters of one of these two models, depending on whether the user confirmed some automatic alignment with ENTER or invalidated it with ESC.

In the following experiments, both models are Gaussians trained on MFCC coefficients. Each Gaussian has two parameters, respectively μ_c and σ_c^2 for the correct alignment model, and μ_b and σ_b^2 for the bad alignment model. The confidence score is the simple average of the log-likelihood ratio over the T frames of the target segment x_1^T:

$$CM(x_1^T) = \frac{1}{T}\sum_{t=1}^{T} \log \frac{\mathcal{N}(x_t; \mu_c, \sigma_c)}{\mathcal{N}(x_t; \mu_b, \sigma_b)}$$

6. Validation

6.1. Efficiency evaluation

This first experiment evaluated the human time required to align a corpus with JTrans and compared it with Transcriber.

The efficiency of JTrans depends on the quality of the audio corpus and the accuracy of the automatic aligner. It was thus evaluated on two types of files, which respectively represent bad (File A) and good (File B) audio quality. File A and File B both have a duration of 15 minutes. Three users evaluated the time required to align each file with a coarse granularity of about one synchronization point per sentence. Note that this granularity was chosen because of the functionalities of Transcriber, but, thanks to its forced alignment capability, JTrans can easily provide finer alignment granularity, at the word or phone level for instance. Table 1 shows the results of this evaluation.

	File A JTrans	File A Transcriber	File B JTrans	File B Transcriber
User1	14'	15'	7'	15'
User2	20'	50'	12'	15'
User3	17'	20'	9'	15'

Table 1: User alignment times for two files with respectively JTrans and Transcriber.

In all cases, the time required to align the files with JTrans is equal to or shorter than with Transcriber. With a good audio quality (File B), the alignment realized with JTrans is always faster than with Transcriber. In this case, more than 90 % of user inputs were confirmation anchors.

We evaluated the alignment accuracy by comparing the timestamps of reference manually-aligned words, with words aligned with JTrans, at the output of stage 2, i.e., just after fast coarse semi-automatic alignment and background batch alignment between anchors. The average difference on File A is less than 0.4s per word.

6.2. Confidence measure evaluation

The performances of the confidence measure were evaluated on File A, by thresholding the confidence measure with a range of different thresholds and plotting the corresponding detection error trade-off (DET) curve in figure 1. Acceptances and rejections respectively correspond to automatic alignments classified as correct and incorrect.

In this experiment, just like in real usage conditions, the confidence measure is updated incrementally as soon as a new user feedback is given. However, to limit the variability of the user's input, a "simulated user" is rather used, which is a program that compares the result of each block-Viterbi with a reference transcription and either sends a confirmation to JTrans or corrects the last proposed word segmentation.

With an equal-error rate (EER) of about 40 %, the proposed confidence measure does not appear very efficient. However,

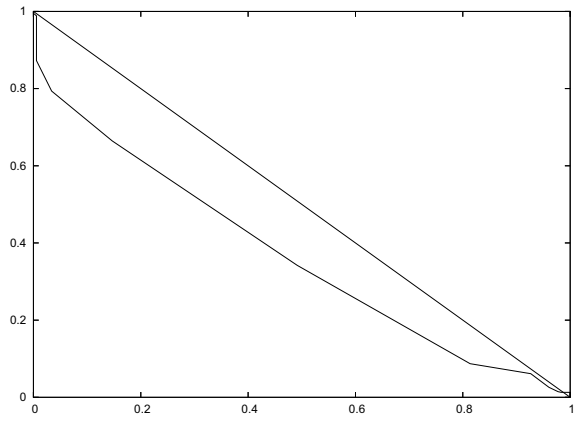

Figure 1: DET curve of the proposed confidence measure on File A. The x-axis represents false rejection rate while the y-axis plots false acceptance rate.

the leftmost part of the curve is the most interesting one: it indicates that all the segments with the lowest confidence score are indeed incorrectly-aligned segments.

Then, in a preliminary experiment, we assessed whether this confidence measure could be used to reduce the number of corrections made by the user. When the confidence measure is below a given threshold, the segment is skipped as explained in section 5. Figure 2 shows the ratio of the number of user confirmations over the number of user corrections, for a range of false rejection rates.

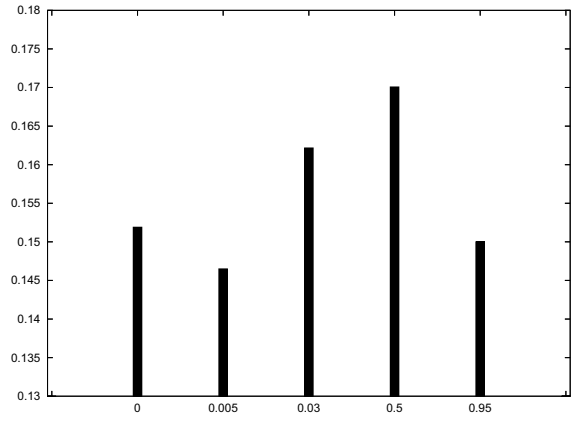

Figure 2: Ratio of number of user confirmations over number of user corrections required to align File A, with the proposed confidence measure and for a range of false rejection rates.

This plot shows that the proposed confidence measure may be used to reduce the number of user corrections required to align a corpus. However, this preliminary experiment should be interpreted with caution and shall be confirmed on a larger set of audio files. The next step shall consist in improving the confidence measure, for instance by introducing information from the speech recognizer itself, estimation of the speaking rate, of the local SNR...

7. Conclusions and future work

JTrans is an OS-independent open-source software that aims at helping researchers in creating and aligning speech corpora. An incremental automatic alignment algorithm that can be controlled on the fly by the user has been developed specifically to greatly facilitate and speed-up the process of speech corpus manipulation. The focus of the software is mainly on the interaction between the user and the automatic algorithms. In addition, a modular design in Java has been chosen to facilitate modifications and improvements of the core libraries, so that they can be easily adapted to other applications. JTrans integrates two original approaches: a guided incremental semi-automatic alignment algorithm, and an alignment confidence measure with incremental adaptation capabilities.

JTrans is designed to be constantly evolving, and it can still be improved in many ways, both regarding the graphical user interface and speech decoding algorithms. The next versions of the software shall include more robust speech recognition algorithms and provide tools for semi-automatic corpus anonymization. The current version of JTrans is distributed with French lexicon and acoustic models, but an English version should be available soon. For now, the user feedback is used both to control the automatic alignment algorithm, and to improve the proposed confidence measure. It shall also be exploited soon to improve the decoding algorithms.

8. Acknowledgments

We would like to thank the Contrat Plan Etat Region MISN TALC `http://misn.loria.fr/talc/index.php` for their support in this project. We also would like to thank the ATILF laboratory researchers who have given us linguistic corpus samples and have helped us in designing and testing JTrans.

9. References

[1] C. Barras, E. Geoffrois, Z. Wu, and M. Liberman, "Transcriber: development and use of a tool for assisting speech corpora production," *Speech Communication*, pp. 5–22, 2001.

[2] P. Martin, "Winpitchpro: A tool for text to speech alignment and prosodic analysis," in *Proc. Speech Prosody 2004*, Nara, Japan, Mar. 2004, pp. 545–548.

[3] P. C. Woodland and S. J. Young, "The HTK continuous speech recogniser," in *Proc. EUROSPEECH*, Berlin, 1993, pp. 2207–2219.

[4] P. J. Moreno, C. Joerg, J.-M. Van Thong, and O. Glickman, "A recursive algorithm for the forced alignment of very long audio segments," in *Proc. ICSLP*, Dec. 1998.

[5] F. de Jong, R. Ordelman, and M. Huijbregts, "Automated speech and audio analysis for semantic access to multimedia," in *Proc. International Conference on Semantic and Digital Media Technologies*, Athens, Greece, Dec. 2006.

[6] A. Haubold and J. R. Kender, "Alignment of speech to highly imperfect text transcriptions," in *Proc. IEEE Conf. on Multimedia and Expo*, July 2007.

[7] T. J. Hazen, "Automatic alignment and error correction of human generated transcripts for long speech recordings," in *Proc. Interspeech*, 2006, pp. 1606–1609.

[8] J.-P. Goldman, "Easyaligner: a semi-automatic phonetic alignment tool under praat," 2007, `http://latlcui.unige.ch/phonetique`.

[9] S. Luz, M. Masoodian, B. Rogers, and C. Deering, "Interface design strategies for computer-assisted speech transcription," in *Proc. the CHISIG Annual Conference on Human-Computer Interaction*, Cairns, Australia, Oct. 2008, pp. 203–210.