A simple and efficient method to align very long speech signals to acoustically imperfect transcriptions

Germán Bordel, Silvia Nieto, Mikel Penagarikano,
Luis Javier Rodríguez-Fuentes, Amparo Varona

GTTS (http://gtts.ehu.es), Department of Electricity and Electronics, ZTF/FCT
University of the Basque Country UPV/EHU, Barrio Sarriena, 48940 Leioa, Spain
german.bordel@ehu.es

Abstract

In the framework of a contract with the Basque Parliament for subtitling the videos of bilingual plenary sessions, which basically consisted of aligning very long (around 3 hours long) audio tracks with syntactically correct but acoustically inaccurate text transcriptions (since all the disfluencies, mistakes, etc. were edited), a very simple and efficient procedure (avoiding the need for language nor lexical models, which was key because of the mix of languages) was developed as a first approach, before trying more complex schemes found in the literature. Since it worked pretty well and the output was quite satisfactory for the intended application, that simple approach was finally chosen. In this paper, we describe the approach in detail and apply it to a widely known annotated dataset (specifically, to the 1997 Hub4 task), to allow the comparison to a reference approach. Results demonstrate that our approach provides only slightly worse segmentations at a much lower computational cost and requiring much fewer resources. Moreover, if the resource to be segmented includes speech in two or more languages and speakers conmute between them at any time, applying a speech recognizer becomes unfeasible in practice, whereas our approach can be still applied with no additional cost.

Index Terms: speech-to-text alignment, automatic video subtitling, multimedia information retrieval, multilingual speech.

1. Introduction

Automatic Speech Recognition (ASR) technology provides high quality results but not enough for supporting completely automatic transcriptions of audio recordings. Therefore, manual speech-to-text transcription is a professional activity that serves to many clients that can only admit perfect transcriptions (e.g. the minutes of parliamentary sessions).

Even more, usually these clients do not need a verbatim transcription of the speech –including all kind of disfluences, mistakes, etc.– but a kind of cleaned transcription that cross the gap between spoken and written language.

Most of the time, these manual transcriptions are a final product that replaces the original source as reference document, but the massive irruption of multimedia access in the last years suggests that synchronizing these texts with their respective sources can add value for many of these clients.

One such application relates to accessibility, in particular the addition of subtitles to the videos offered by companies and organizations through the Internet —like, in our case, the Basque Parliament [1]—. Another obvious application is information retrieval, where the actual mechanisms allowing access to these textual resources can be extended to the source videos that can allow a more comprehensive understanding (attending at the non-verbal aspects of the communication process).

The alignment task is not difficult if forced alignment at word level can be carried out by constraining the recognizer to match the sequence of words given by the text, and allowing some mismatch between speech and text to cope with imperfect transcripts or alternative pronunciations [2] [3] (an interesting analysis of editions and mistakes introduced by human transcriptors can be found in [4]). In any case, the search space must be small enough to get to good results.

The problem arises when very long signals have to be aligned. State-of-the-art technology can just deal with few minutes of speech. The common solution to this problem was given in [5], where the strategy consisted on analyzing the output of an ASR system to look for sequences of words that match the text (called anchors), and considering these points as split points to reduce the length of the problem. This was done recursively until a forced alignment could be done for each chunk of speech. The reported results were really good. In fact the alignment was done in a forced manner, and as long as there were no errors in choosing the anchors, the precision was that of the ASR system.

The method proposed in [5] implicitly considers that finding a more direct match between both sequences would not work properly. That direct match was what we wanted to test in advance before implementing the method in [5]. In this paper we present the very simple and efficient procedure that consists on obtaining a phone decoding of the speech signal and align it to the phonetic transcription of the reference test. An approach in this direction —relying on phonetical alignment— was presented in [6] but it was aimed to a different objective: finding correct words in a highly imperfect transcription. This simple approach has no heuristics and gives results that are not so far from the more complex procedures. In any case, works that study the quality of alignments in terms of the duration of words, the effects of insertions and deletions, the scores of the recognizer, etc. like [7] are aimed to better even the forced alignment under adverse conditions, and a direct solution like ours can also benefit from it.

The rest of the paper is organized as follows. In Section 2 the alignment method is described in detail. In section 3 we present the video subtitling application for the Basque Parliament and results of the proposed approach on the 1997 HUB4 corpus [8]. Finally section 4 discusses inner workings of the alignment procedure and some improvements currently under consideration.

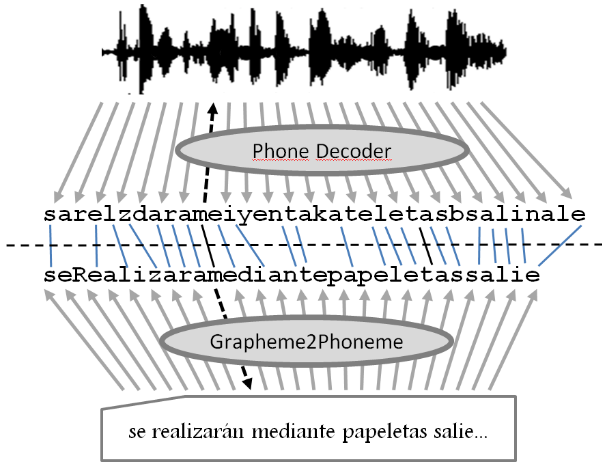
Phone Decoder
sarelzdarameiyentakateletasbsalinale
seRealizaramediantepapeletassalie
Grapheme2Phoneme
se realizarán mediante papeletas salie...

Figure 1: *The speech-to-text alignment process is defined as the alignment between a recognized phone transcription and the phone transcription of the reference text*

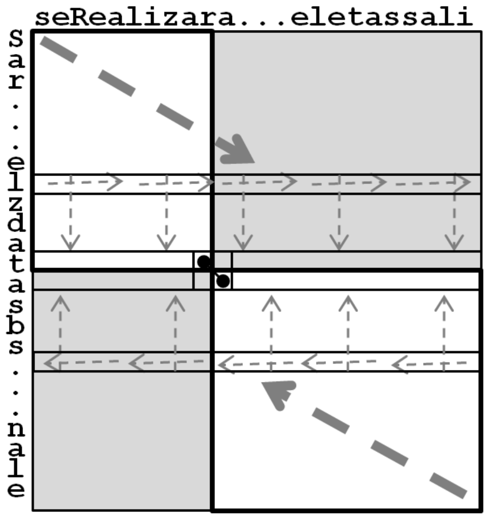

Figure 2: *The Hirschberg algorithm allows the optimal alignment of two very long symbol sequences*

2. The speech-to-text alignment method

Given that speech and text can be translated to a common symbolic representation (words, phonemes or whatever other acoustic units) the most straightforward approach to the synchronization of both streams would map them into this common representation, and relate positions in the original sources by their correspondence to the symbol stream (see Figure 1).

We chose the phonetic transcription because a small vocabulary size and a small granularity would help the alignment between both sequences. This decision assumes that phone decoding is performed without any language nor phonotactic model, so that this part of the system is language independent (given that the phoneme set will be able to cover all the languages appearing in the speech stream). Nevertheless, note that the language model is present in the system as we count on the exact phone sequence given by the text transcription. (The use of a transcription based bigram model in the speech part was studied, but did not show any significant improvement)

The grapheme-to-phoneme conversion is the easy part, provided that a good dictionary is available. In section 2.2 we comment some details and particularities related to this part of the process.

Once both phone sequences are available, the symbol alignment procedure will find the best match, the quality of the result depending basically on the procedures used to translate both streams to the common phonetic representation.

2.1. From speech to phones

Using a phone decoder allowed our system to cope with the mixed use of Spanish and Basque practiced in the Basque Parliament just by training a set of phonemes covering both languages. For the HUB4 tests, an English decoder was needed, and a 40 phone set of models was estimated on the TIMIT database [9] and then re-trained on the Wall Street Journal database [10]. In both cases, left-to-right non-contextual continuous Hidden Markov Models, with three looped states and 64 Gaussian distributions per state, were used as acoustic models.

The original audio streams were downsampled to 16 KHz, storing them in PCM format, 16 bit per sample. The resulting signals were processed to obtain a feature vector every 10 ms using a 25 ms Hamming window, first order preemphasis (0.97 coefficient), and a 26-channel Mel-scale filterbank, resulting 39-dimensional feature vectors, consisting of 12-order Mel Frequency Cepstral Coefficients (MFCC) plus energy, and their delta and delta-delta coefficients.

2.2. From text to phones

Given that our system is able to cope with a mixed use of Spanish and Basque, transcribing text to phones in both languages is a key feature. We built a multilingual transcriber based on two monolingual transcribers, each composed by a dictionary, a set of transcription rules and a ”number-and-symbols to text converter” (for numbers, currencies, percent, degrees, abbreviations, etc). There is also a third transcriber with just a dictionary to cover all the words out of the vocabulary of both languages. For each word, the transcriber asks the three subsystems if that word it is contained in their dictionaries and, if it receives a unique positive response, the transcription is accepted. In any other cases it analyzes the context to determine the language being used and accepts the word in the corresponding dictionary if a multiple match was the case, or asks the corresponding subsystem for a rule based transcription. These new rule-based transcriptions are added to the corresponding dictionary and reported to be supervised. This mechanism produces transcriptions based on verified dictionaries that grow incrementally, and at the same time acts as a misspelling detector and allows the refinement of rules.

For the HUB4 tests we just used the CMU pronouncing dictionary [11].

2.3. Alignment of very long sequences

The optimal solution to the alignment of two symbol sequences is given by the Needleman-Wunsh algorithm [12]. Being S_n and S_m two sequences of symbols with lengths n and m respectively, it basically consists on filling a $n \times m$ matrix from top to bottom with accumulated minimized edition costs (Levenshtein's distance) and track back the lowest values path from the bottom right corner to top left corner.

Obviously this method is prohibitive for very long sequences due to the matrix memory allocation (our typical phonetic sequence is about 100,000 symbols long for a 3 hours signal, so considering a 4 byte integer per cell, we would need roughly 40GB for the matrix). In this case, when memory availability is a constraint, we can still use the divide and conquer version known as Hirshberg algorithm [13]. Figure 2 outlines the process: it finds the columns where the optimal path crosses the central rows by doing all the matrix calculations but storing only one row that goes half matrix forward from the start, and one row that goes backward from the end; once this point is located, the problem splits into the application of the same procedure to the two submatrices in the principal diagonal. The recursion can reach its base case when the amount of memory needed to apply the non-recursive algorithm can be allocated. This algorithm reduces dramatically the required memory at a cost in processing time that typically supposes less than ×2 factor, but it is easily parallelizable and this impact can be reduced even for a typical desktop computer (less than 1 minute for a 3 hours signal in an 8 threaded Intel i7 2600).

3. Alignment accuracy results

3.1. Application to the Basque Parliament videos

The above system is being used since September 2010 to subtitle the Basque Parliament plenary sessions videos served in the web, totaling 80 sessions and 407 hours of video so far. After the alignment is done, and following a certain set of rules related to lengths, times and punctuation, the synchronized text stream is split in chunks suited to captioning. In consequence, only the first phoneme of each chunk is taken into account to synchronize text and speech, and errors are perceived by web users as these chunks being presented with some advance or delay. This perception admits different tolerances depending on the flow of the speech but, in general, the task is not very demanding: for a continuous speech, a 0.5 second deviation is not too much; after a long silence, when captions blank, the next caption admits a considerable higher deviation in advance but not in delay.

From the first moment, after a manual supervision for the first sessions —and given the quality of the inputs— it was considered that the generated subtitles were suitable and the system started to be used in production. No more than 8 points are manually repositioned for each session, and these errors obeys to an easy to locate pattern of background voice in silence (section 4 provides more details related to this issue).

3.2. Application to the 1997 Hub4

The proposed method was good enough for the subtitling task on parliamentary sessions. Then we tested it on the 1997 HUB4 database, in order to assess its behavior when confronted to a different kind of resource.

The 1997 Hub4 database is composed of about 3 hours of transcribed broadcast audio, annotated according to 6 categories for the acoustic conditions plus a seventh ”other” category and a set of ”unclassified” segments. We discarded the last two subsets (amounting to 8.5% phonemes) and considered only the six following categories: F0-clean speech, F1-spontaneous, F2-telephone, F3-with background music, F4-degraded acoustics, and F5-from non natives. Their proportions are shown in Figure 3.

To measure the alignment accuracy, the deviation of the starting point of each word from the precalculated ground truth position was computed. To obtain these reference positions, a sentence by sentence forced alignment was performed using a recognizer whose models where closely adapted to the HUB4 database.

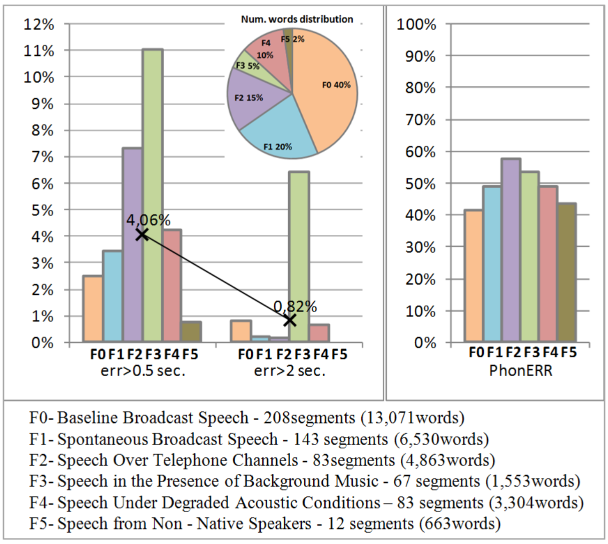

Figure 3: *1997 HUB4 structure, phonetic decoding error rates and alignment results. The alignment is evaluated according to two different criteria: an error is counted when the word start differs from the reference more than 0.5 or 2 seconds. In the first case the average alignment error is around 4%; in the second it reduces to 0.8%. Different acoustic conditions affect differently the alignment accuracy.*

Results are summarized in Figure 3. The phonetic decoder presents error rates in the 40%-60% range depending on the acoustic conditions. As in [5], we evaluate the alignment procedure by classifying as errors all the words starting more than 0.5 seconds and more than 2.0 seconds apart from the reference time. The average error figures are approximately 4% and 0.8% respectively. It is interesting to note that the alignment errors does not fit the same pattern than the decoder error rates. Background music is the worst condition making a significant number of words to misplace a significant amount of time. Speech over telephone channels is the worst condition for the decoder but even it presents a quite high error rate for the alignment at 0.5 seconds condition, it is less than for background music and, what is more interesting, the error rate for 2 seconds is very low. This suggest that the error distribution affects in some manner (few well recognized phones can act as achors even when the PhonERR is high, and give place to better alignments than a lower but more uniform error distribution).

4. Discussion and future work

It is evident that a better phonetic decoder will improve the results shown above. For any given task, better phonetic models can be used, by starting with the general models and then adapting them to the particular resources to be aligned.

Another idea to explore is the substitution of the kernel corresponding to the Levenshtein's distance by a more informative kernel about the decoder confusion probabilities.

In any case, there is an interesting work to do in the characterization of the information that can be extracted from the alignment path, that can be later used to automatically reconsider some word synchronizations. Figure 4 represents a section of an alignment path. Interpreting it as is explained in

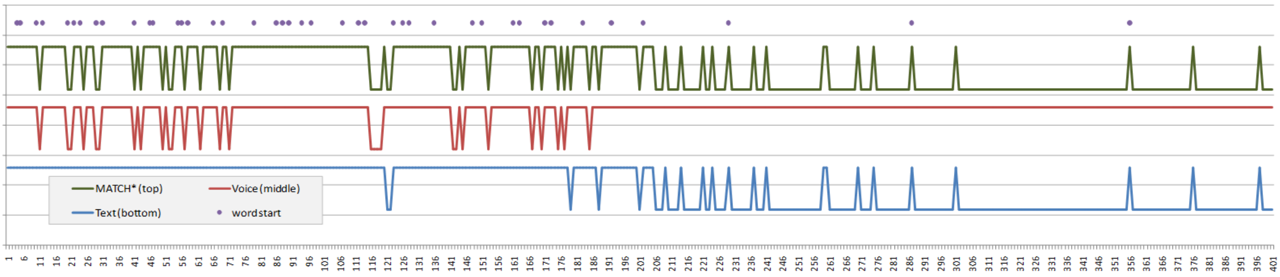

Figure 4: *Patterns found in the trellis. This segment represents a section of an alignment path: the bottom line is up for phonemes in the text, whereas the middle line is up for phonemes in the recognized speech (so, points down are decoder insertions for the bottom line and deletions for the middle one). The top line is the AND function over the previous couple, that is, upper state corresponds to decoder matches and substitutions, and lower state corresponds to insertions and deletions. Two different halves can be identified: the left one corresponds to a correct alignment, whereas the right one tell us that there is a lack of transcription for some utterances. At the left side, words are detected at distances that are basically in accordance to their phonetic lengths (upper dots); at the right side, a long run of decoded phonemes (middle line up) matches to few text phonemes (bottom line mostly down): the words in the text are sparsely matched with phonemes from the non-transcribed speech. From the half correctly aligned, we also find that the phone decoder is applying an excess insertion penalty (there are more deletions than insertions in the recognized phone sequence)*

the caption, we see that there are two patterns giving us hints about what is going on. When the alignment is working right, the words are associated to a path length that is close to its phoneme length. The relation between matches and substitutions and the time span for each word give us more information about the probability of being a perfect match. Places where there is no reference transcription can be detected as long runs of phoneme insertions, that is, as words spanning in excess through the alignment path. The opposite situation (extra text), rarely present in manual transcriptions, will generate long runs in the other axis, that is, log runs of deletions. Both problems generate border effects that should be corrected. These border effects depend on the severity of the problem (the number inserted or deleted words) but nevertheless the attraction or repulsion effect that this regions have on the recognized sequence of phonemes is somehow smoothed by the need that they match a given sequence.

These considerations give way to an opposite mechanism to that established by [5]: given that most of the alignment is right (most of it acts as anchor), it allows us to focus on the problematic areas to isolate the translation mistakes and correct the border effects by means of forced alignment.

5. References

[1] G. Bordel, S. Nieto, M. Penagarikano, L. J. Rodriguez Fuentes, and A. Varona, “Automatic subtitling of the basque parliament plenary sessions videos,” in *Twelfth Annual Conference of the International Speech Communication Association, INTERSPEECH*, 2011, pp. 1613–1616.

[2] J. Vonwiller, C. Cleirigh, H. Garsden, K. Kumpf, R. Mountstephens, and I. Rogers, “The development and application of an accurate and flexible automatic aligner,” *The International Journal of Speech Technology*, vol. 1, no. 2, pp. 151–160, 1997.

[3] P. Moreno and C. Alberti, “A factor automaton approach for the forced alignment of long speech recordings,” in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2009, pp. 4869 –4872.

[4] T. Hazen, “Automatic alignment and error correction of human generated transcripts for long speech recordings,” in *Ninth International Conference on Spoken Language Processing, Interspeech-ICSLP*, 2006.

[5] P. Moreno, C. Joerg, J. Thong, and O. Glickman, “A recursive algorithm for the forced alignment of very long audio segments,” in *Fifth International Conference on Spoken Language Processing*, 1998.

[6] A. Haubold and J. Kender, “Alignment of speech to highly imperfect text transcriptions,” in *Multimedia and Expo, 2007 IEEE International Conference on*. IEEE, 2007, pp. 224–227.

[7] R. Das, J. Izak, J. Yuan, and M. Liberman, “Forced alignment under adverse conditions,” *University of Pennsylvania, CIS Dept. Senior Design Project Report*, 2010.

[8] D. Graff, J. Fiscus, and J. Garofolo, “1997 hub4 english evaluation speech and transcripts,” Linguistic Data Consortium, Philadelphia, 2002.

[9] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, “Timit acoustic-phonetic continuous speech corpus,” Linguistic Data Consortium, Philadelphia, 1993.

[10] J. S. Garofolo, D. Graff, D. Paul, and D. S. Pallett, “Csr-i (wsj0) complete,” Linguistic Data Consortium, Philadelphia, 2007.

[11] R. Weide, “The carnegie mellon pronouncing dictionary [cmudict. 0.6],” Carnegie Mellon University, 2005. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[12] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443 – 453, 1970.

[13] D. Hirschberg, “A linear space algorithm for computing maximal common subsequences,” *Communications of the ACM*, vol. 18, no. 6, pp. 341–343, 1975.