10.12871/clicit2014224

The SPPAS participation to Evalita 2014

Brigitte Bigi
Laboratoire Parole et Langage, CNRS, Aix-Marseille Université,
5 avenue Pasteur, BP80975, 13604 Aix-en-Provence France
brigitte.bigi@lpl-aix.fr

Abstract

English. SPPAS is a tool to automatically produce annotations which includes utterance, word, syllabic and phonemic segmentation from a recorded speech sound and its transcription. This paper describes the participation of SPPAS in evaluations related to the “Forced Alignment on Children Speech” task of Evalita 2014. SPPAS is a ”user-friendly” software mainly dedicated to Linguists and open source.

Italiano. *SPPAS è uno strumento in grado di produrre automaticamente annotazioni a livello di parola, sillaba e fonema a partire da una forma d'onda e dalla sua corrispondente trascrizione ortografica. Questo articolo descrive la partecipazione di SPPAS nelle valutazioni relative al task Forced Alignment on Children Speech (allineamento forzato su parlato infantile) di Evalita 2014. SPPAS è un software ”open source”, è molto semplice da utilizzare ed è particolarmente indicato all'uso da parte di linguisti.*

1 Introduction

EVALITA is an initiative devoted to the evaluation of Natural Language Processing and Speech tools for Italian[1]. In Evalita 2011 the “Forced Alignment on Spontaneous Speech” task was added. Then, in 2014, this task is evolving to “Forced Alignment on Children Speech” (FACS). Nevertheless, as in 2011, systems were required to align a set of audio sequences to the provided relative transcriptions. Forced-aligment (also called phonetic segmentation) is the process of aligning speech with its corresponding transcription at the phone level. The alignment problem consists in a time-matching between a given speech unit along with a phonetic representation of the unit. The goal is to generate an alignment between the speech signal and its phonetic representation. Speech alignment requires an acoustic model in order to align speech. An acoustic model is a file that contains statistical representations of each of the distinct sounds of one language. Each phoneme is represented by one of these statistical representations.

After Evalita 2011 (Bigi, 2012), this paper presents the SPPAS participation to the FACS task. The training procedure and the corpus we used during the development phase to provide a new acoustic model are described.

2 Acoustic models: Training procedure

Phoneme alignment is the task of proper positioning of a sequence of phonemes in relation to a corresponding continuous speech signal. In the alignment problem, we are given a speech utterance along with a given phonetic representation of the utterance. Our goal is to generate an alignment between the speech signal and the phonetic representation.

SPPAS (Bigi, 2011) is based on the Julius Speech Recognition Engine (Nagoya Institute of Technology, 2010). Julius was designed for dictation applications, and the Julius distribution only includes Japanese acoustic models. However since it can use acoustic models trained using the Hidden Markov Toolkik (HTK) (Young and Young, 1994), it can also be used in any other language.

Acoustic models were then trained with HTK using the training corpus of speech, previously segmented in utterances, phonetized and automatically time-aligned. The trained models are Hidden Markov models (HMMs). Typically, the HMM states are modeled by Gaussian mixture densities whose parameters are estimated using an expecta-

[1] http://www.evalita.it/

tion maximization procedure. The outcome of this training procedure is dependent on the availability of accurately annotated data and on good initialization. Acoustic models were trained from 16 bits, 16000 hz wav files. The Mel-frequency cepstrum coefficients (MFCC) along with their first and second derivatives were extracted from the speech in the standard way (MFCC_D_N_Z_0).

The training procedure is based on the VoxForge tutorial[2], except that which from VoxForge uses word transcription as input. Instead, we took as input the proposed phonetized transcription, with or without using the phonetic time-alignment. This procedure is based on 3 main steps: 1/ data preparation, 2/ monophones generation then 3/ triphones generation.

Step 1 is the data preparation. It establishes the list of phonemes, plus fillers, silence and short pauses. It converts the input data into the HTK-specific data format (MLF files). It codes the audio data, also called "parameterizing the raw speech waveforms into sequences of feature vectors" (i.e. convert from wav to MFCC format), using "HCopy" command.

Step 2 is the monophones generation. In order to create a HMM definition, it is first necessary to produce a prototype definition. The function of a prototype definition is to describe the form and topology of the HMM, the actual numbers used in the definition are not important. Having set up an appropriate prototype, a HMM can be initialized by both methods:

- create a flat start monophones model, a prototype trained from phonetized data, and copied for each phoneme (using "HCompV" command). It reads in a prototype HMM definition and some training data and outputs a new definition in which every mean and covariance is equal to the global speech mean and covariance.

- create a prototype for each phoneme using time-aligned data (using "Hinit" command). Firstly, the Viterbi algorithm is used to find the most likely state sequence corresponding to each training example, then the HMM parameters are estimated. As a side-effect of finding the Viterbi state alignment, the log likelihood of the training data can be computed. Hence, the whole estimation process can be repeated until no further increase in likelihood is obtained.

In our script, we train the flat start model and we fall back on this model for each phoneme that fails to be trained with Hinit (if there are not enough occurrences). This first model is re-estimated using the MFCC files to create a new model, using "HERest". Then, it fixes the "sp" model from the "sil" model by extracting only 3 states of the initial 5-states model. Finally, this monophone model is re-estimated using the MFCC files and the phonetized data.

Step 3 creates tied-state triphones from monophones and from some language specificities defined by means of a configuration file. This file summarizes Italian phonemic information as for example the list of vowels, liquids, fricatives, nasals or stop. We created manually this resource, and distribute it on-demand.

3 Corpus description

The training set is made of children recorded while reading some text and is available in the form of time-aligned sentences (one file per sentence). The result of an automatic word segmentation and phoneme segmentation is also available. In addition to the Child corpus, the data of Evalita 2011 were also distributed. Some other data were also collected in the scope of this study: a/ 5300 isolated pluri-syllabic tokens of Italian children, with various recording conditions (often with a poor audio quality); b/ read speech of 41 speakers, recorded at Torino (all speakers are reading the same text), the total duration is 31275.8 seconds. This corpus is available at: http://sldr.org/ortolang-000894

In order to create a development set, some files were randomly picked up of the Child set and manually time-aligned by the author (not phonetician), using Praat with the help of the spectrogram. Then 134 files were annotated, with a duration of 888.77 seconds. It is to be noticed that the phonetization was not changed, only the time-alignments were modified. The time spent to correct the automatic alignments was about 9-10 hours. This development corpus contains 196 silences, 60 fillers, 326 /a/, 218 /e/, 218 /o/ and 192 /i/. For this corpus, 2529 boundaries have to be fixed by the system.

In the evaluations, we propose detailed alignment performances depending on the delta range

[2] http://www.voxforge.org

between the automatic and the reference alignments, using the time-localization of the end-bound of each phoneme.

4 Experiment 1: time-aligned data is good data?

In this experiment, we try to fix which amount of data is required for the initial model of step 2. Only the Child corpus is used: the phonetization of the whole corpus is used in all other stages of the training procedure, and time-aligned data are used only to train the initial model. Results are reported in Figure 1. We can observe that, for this stage of the training procedure, 30 seconds of automatic time-aligned speech are the strict minimum that must be used. It seems that 5 minutes are a good compromise. Then, the data used for this initial model are now fixed (they will not be changed in further experiments): the speech duration for the initial model is 302.72 seconds.

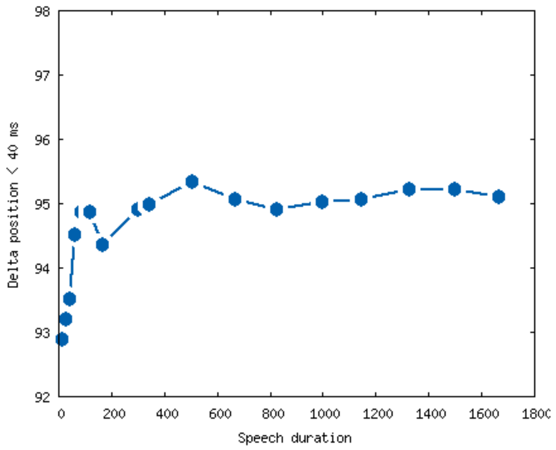

Figure 1: Experiment 1. Results depending of the amount of speech data to train the initial model.

5 Experiment 2: more data is good data?

By fixing the initial model as mentioned in the previous section, we will now evaluate the results while changing the amount of phonetized data (still in step 2, to train the monophones). In this experiment, only the Child corpus is used too. Results are reported in Figure 2. We can observe that from 3 to 10 minutes of data, the differences are very slights, withal we can conclude that more data is good data. However, the differences are not significant for experiments with more than 10 minutes of phonetized speech.

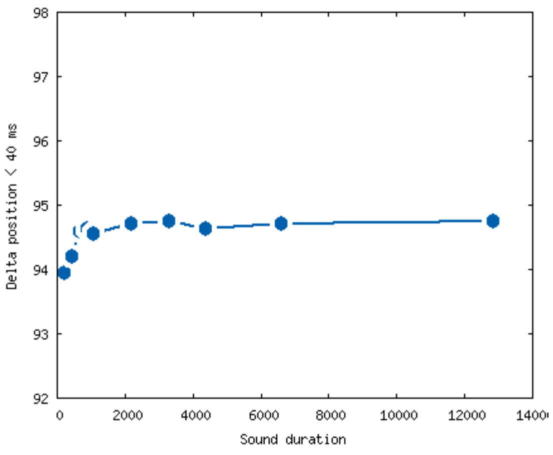

Figure 2: Experiment 2. Results depending of the amount of phonetized speech data.

6 Experiment 3: other data is good data?

We added the data from the CLIPS, distributed by the organizers and then our own data.

Results are reported in Table 1.

Our conclusion is that more data is not good data, and we decided the following: a/ to remove our children corpus of the training data set; b/ to use triphones; c/ to add 5 minutes of time-aligned data of the CLIPS corpus to train the initial model.

7 Final models

We finally trained 3 models by choosing data sets on the basis of the experiments described in the previous sections. The ”Closed A” model was trained using Child and CLIPS corpora, the ”Closed B” model using only Child and the ”Open” model using both Child and CLIPS corpora plus a free corpus available on the web (previously named ”read-Torino”). Results on the development corpus, within a delta of 40 ms, are:

- ”Closed A” 2400 (94.90%)
- ”Closed B” 2406 (95.14%)
- ”Open” 2389 (94.46%)

Figure 3 show detailed results on vowels of the ”Open” model, distributed in SPPAS-1.6.1.

8 Conclusion

During this evaluation campaign, we asked 3 questions and answered within the FACS context. We asked if ”time-aligned data is good data?” and

Model	Monophones		Triphones	
Phonetized Corpus	# Corr	%Corr	# Corr	%Corr
Only Child	2396	94.74	2404	95.06
Child + dialog-CLIPS	2390	94.50	2395	94.70
Child + read-Torino	2394	94.66		
Child + read-children	2381	94.15		
Child + dialog-CLIPS + read-Torino	2390	94.50	2389	94.46
Child + dialog-CLIPS + read-Torino + read-children	2380	94.11	2362	93.40

Table 1: Results of experiment 3, in a delta less than 40ms.

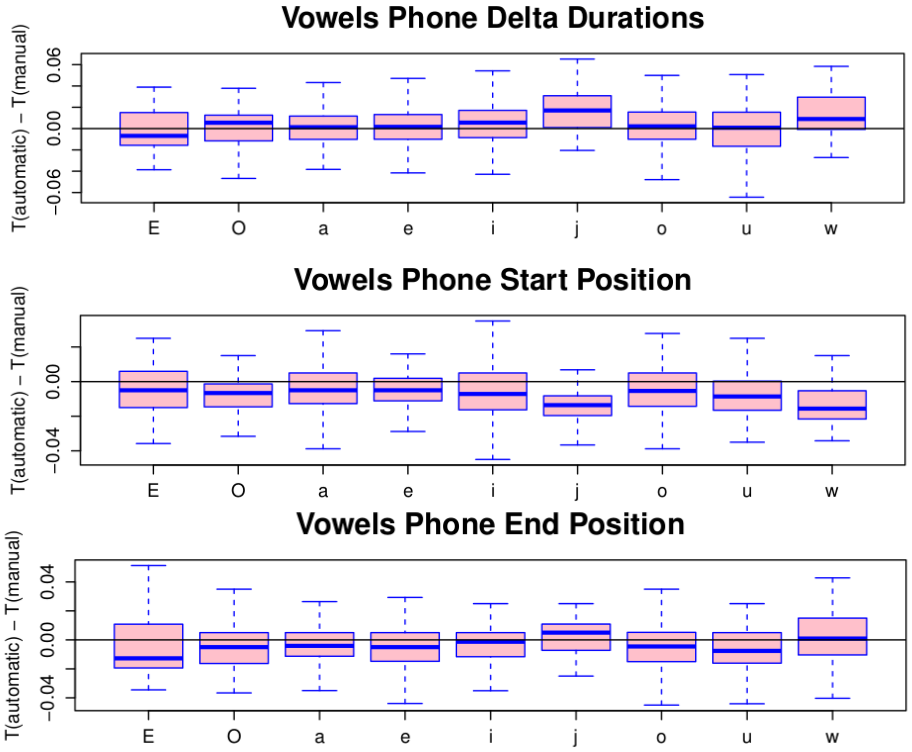

Figure 3: Results on vowels of the ”Open” model.

found that 5 minutes are a good amount of time-aligned data to train the initial model. We asked if ”more data is good data?” and found that at least 10 minutes of phonetized data are required (with more data, the benefits are very slights). We finally asked if ”other data is good data?” and found that the answer is no, a dedicated system is better than a general one (which is not surprisingly).

Acknowledgments

This work has been carried out thanks to the support of the A*MIDEX project (ANR-11-IDEX-0001-02) funded by the ”Investissements d’Avenir” French Government program, managed by the French National Research Agency (ANR).
URL of the project:
http://variamu.hypotheses.org

References

[Bigi2011] B. Bigi. 2011. SPPAS - Automatic Annotation of Speech, http://www.lpl-aix.fr/∼bigi/sppas/.

[Bigi2012] B. Bigi. 2012. The sppas participation to evalita 2011. *Working Notes of EVALITA 2011*.

[Nagoya Institute of Technology2010] Nagoya Institute of Technology. 2010. Open-source large vocabulary csr engine julius, rev. 4.1.5.

[Young and Young1994] S.J. Young and Sj Young. 1994. The htk hidden markov model toolkit: Design and philosophy. *Entropic Cambridge Research Laboratory, Ltd*, 2:2–44.