The Automatic Search for Sounding Segments of SPPAS: Application to Cheese! Corpus

Brigitte Bigi, Béatrice Priego-Valverde

▶ **To cite this version:**

Brigitte Bigi, Béatrice Priego-Valverde. The Automatic Search for Sounding Segments of SPPAS: Application to Cheese! Corpus. Human Language Technology. Challenges for Computer Science and Linguistics., 13212, Springer, pp.16 - 27, 2022, Lecture Notes in Computer Science, 978-3-031-05328-3. 10.1007/978-3-031-05328-3_2 . hal-03697808

HAL Id: hal-03697808
https://hal.science/hal-03697808

Submitted on 23 Sep 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

The automatic search for sounding segments of SPPAS: application to Cheese! corpus

Brigitte Bigi and Béatrice Priego-Valverde

LPL, CNRS, Aix-Marseille Univ.
5, avenue Pasteur, 13100 Aix-en-Provence, France
brigitte.bigi@lpl-aix.fr, beatrice.priego-valverde@univ-amu.fr

Abstract. The development of corpora inevitably involves the need for segmentation. For most of the corpora, the first segmentation to operate consist in determining silences vs Inter-Pausal Units - IPUs, i.e. sounding segments. This paper presents the "Search for IPUs" feature included in SPPAS - the automatic annotation and analysis of speech software tool distributed under the terms of public licenses. Particularly, this paper is focusing on its evaluation on Cheese! corpus, a corpus of reading then conversational speech between two participants. The paper reports the number of manual actions which was performed manually by the annotators in order to obtain the expected segmentation: add new IPUs, ignore irrelevant ones, split an IPU, merge two consecutive ones and move boundaries. The evaluation shows that the proposed fully automatic method is relevant.

Keywords: speech, IPUs, segmentation, silence, corpus, conversation

1 Introduction

Corpus Linguistics is a Computational Linguistics field which aims to study the language as expressed in corpora. Nowadays, Annotation, Abstraction and Analysis (the 3A from [9]) is a common perspective in this field. Annotation consists of the application of a scheme to recordings (text, audio, video, ...). Abstraction is the mapping of terms in the scheme to terms in a theoretically motivated model or dataset. Analysis consists of statistically probing, manipulating and generalizing from the dataset. Within these definitions, this paper focuses on *annotation* which "can be defined as the practice of adding interpretative, linguistic information to an electronic corpus of spoken and/or written language data. 'Annotation' can also refer to the end-product of this process" [6]. Annotating corpora is of crucial importance in Corpus Linguistics. More and more annotated corpora are now available, and so are tools to annotate automatically and/or manually. Large multimodal corpora are now annotated with detailed information at various linguistic levels. The temporal information makes it possible to describe behaviour or actions of different subjects that happen at the same time.

An orthographic transcription is often the minimum requirement for a speech corpus. It is often at the top of the annotation procedure, and it is the entry point for most of the other annotations and analysis. However, in the specific annotation context of a multimodal corpus, the time synchronization of the transcription is of crucial importance.

In recent years, SPPAS [2] has been developped by the first author to automatically produce time-aligned annotations and to analyze annotated data. The SPPAS software tool is multi-platform (Linux, MacOS and Windows) and open source issued under the terms of the GNU General Public License. It is specifically designed to be used directly by linguists. As a main functionality, SPPAS allows to perform all the automatic annotations that are required to obtain the speech segmentation at the word and phoneme level of a recorded speech audio and its orthographic transcription [3].

Figure 1 describes the full process of this method in order to annotate a multimodal corpus and to get time-synchronized annotations, including the speech segmentation. The audio signal is analyzed at the top-level of this procedure in order to search for the Inter-Pausal Units. IPUs are defined as sounding segments surrounded by silent pauses of more than X ms. They are time-aligned on the speech signal. IPUs are widely used for large corpora in order to facilitate speech alignments and for the analyses of speech, like prosody in (Peshkov et al., 2012). The orthographic transcription is performed manually at the second stage and is done inside the IPUs.

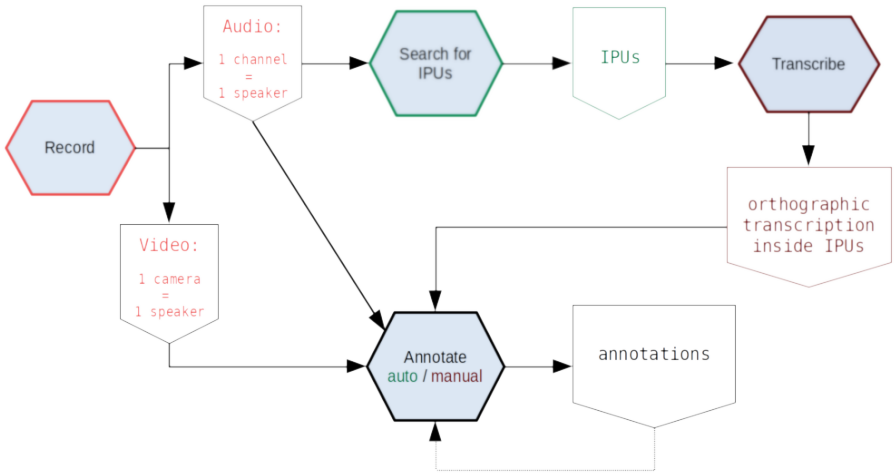

Fig. 1. Annotation process method

We applied this annotation method to the conversational French multimodal corpus 'Cheese!' [7,8]. "Cheese!" is made of 11 face-to-face dyadic interactions, lasting around 15 min each.

This paper presents the automatic annotation "Search for IPUs" of the SPPAS software tool. Given a speech recording of Cheese!, the goal was to generate an annotation file in which the sounding segments between silences are marked. This paper describes the method we propose for a fully automatic search for IPUs. The second section of this paper briefly presents Cheese!. We then propose a user-oriented evaluation method based on the amounts of manual interventions which were required to obtain the final IPUs annotation. Finally, the results are presented as appropriate for the intended use of the task and accompagnied by a qualitative discussion about the errors.

2 The method to search automatically for IPUs

2.1 Algorithm and settings

The search for IPUs is performed on a recorded audio file made of only one channel (mono) and lossless.

Evaluation of a threshold value:
The method is using the Root-Mean-Square (rms), a measure of the power in an audio signal. The rms can be evaluated on the whole audio channel or on a fragment of n of its samples. It is estimated as follow:

$$rms = \sqrt{\frac{\sum_i^n (S_i^2)}{n}} \tag{1}$$

At a first stage, the search for IPUs method estimates the rms value of each fragment of the audio channel. The duration of these fragment windows is fixed by default to 20 ms and can be configured by the user.

The statistical distribution of the obtained rms values is then analyzed. Let min be the minimum rms value, μ the mean and σ the coefficient of variation. Actually, if the audio is not as good as it is expected, the detected outliers values are replaced by μ and the analysis is performed on this new normalized distribution.

A threshold value Θ *is fixed automatically* from the obtained statistical distribution. The estimation of Θ is of great importance: it is used to decide whether each window of the audio signal is a silence or a sounding segment:

- silence: $rms < \Theta$;
- sounding segment: $rms \geq \Theta$

We fixed the value of Θ as follow:

$$\Theta = min + \mu - \delta \tag{2}$$

The δ value of equation (2) was fixed to 1.5σ. All these parameters were empirically fixed by the author of SPPAS from her past experience on several corpora and from the feedback of the users.

It has to be noticed that Θ strongly depends on the quality of the recording. The value fixed automatically may not be appropriate on some recordings, particularly if they are of low-quality. By default it is estimated automatically but it can optionnally be turned off and the user can fix it manually.

Get silence vs sounding fragment intervals:
The rms of each fragment window is compared to Θ and the windows below and above the threshold are identified respectively as silence and sounding. The neighboring silent and neighboring sounding windows are grouped into intervals. At this stage, we then have identified intervals of silences and intervals with "sounds". However, their duration can be very short and a filtering/grouping system must be applied to get intervals of a significant duration.

Because the focus is on the sounding segments, the resulting silent intervals with a too small duration are removed first (see the discussion section below). The minimum duration is fixed to 200 ms by default. This is relevant for French, however it should be changed to 250 ms for English language. This difference is mainly due to the English voiceless velar plosive /k/ in which the silence before the plosion could be longest than the duration fixed by default.

Construction of the IPUs:
The next step of the algorithm starts by re-grouping neighboring sounding intervals that resulted because of the removal of the too short silences. At this stage, the new resulting sounding intervals with a too small duration are removed. This minimum duration is fixed to 300 ms by default. This value has to be adapted to the recording conditions and the speech style: in read speech of isolated words, it has to be lowered (200 ms for example), in read speech of sentences it could be higher but it's not necessary to increase it too much. In spontaneous speech like in conversationnal speech, it has to be lowered mainly because of some isolated feedback items, often mono-syllabic, like 'mh' or 'ah'.

The algorithm finally re-groups neighboring silent intervals that resulted because of the removal of the too short sounding ones. It then make the Inter-Pausal Units it searched for. Silent intervals are marked with the symbol '#' and IPUs are marked with 'ipus_' followed by its number.

The algorithm and its settings in a nutshell:

1. fix a window length to estimate rms (default is 20 ms);
2. estimate rms values on the windows and their statistical distribution;
3. fix automatically a threshold value to mark windows as sounding or silent - this value can be fixed manually if necessary;
4. fix a minimum duration for silences and remove too short silent intervals (default is 200 ms);
5. fix a minimum duration for IPUs and remove too short sounding intervals (default is 300 ms);
6. tag the resulting intervals with # or ipu_i.

2.2 Optional settings

From our past experience of distributing this tool, we received users' feedback. They allowed us to improve the values to be fixed by default this paper mentioned in the previous section. These feedbacks also resulted in adding the following two options:

- move systematically the boundary of the begin of all IPUs (default is 20 ms);
- move systematically the boundary of the end of all IPUs (default is 20 ms).

A duration must be fixed to each of the two options: a positive value implies to increase the duration of the IPUs and a negative to reduce them. The motivation behind these options comes from the need to never miss aso unding part. To illustrate how this might work, one of the users fixed the first value to 100 ms because his study focused on the plosives at the beginning of isolated words.

Figure 2 shows the full list of required parameters and optional settings when using the Graphical User Interface. The same parameters have to be fixed when using the Command-Line User Interface named `searchipus.py`.

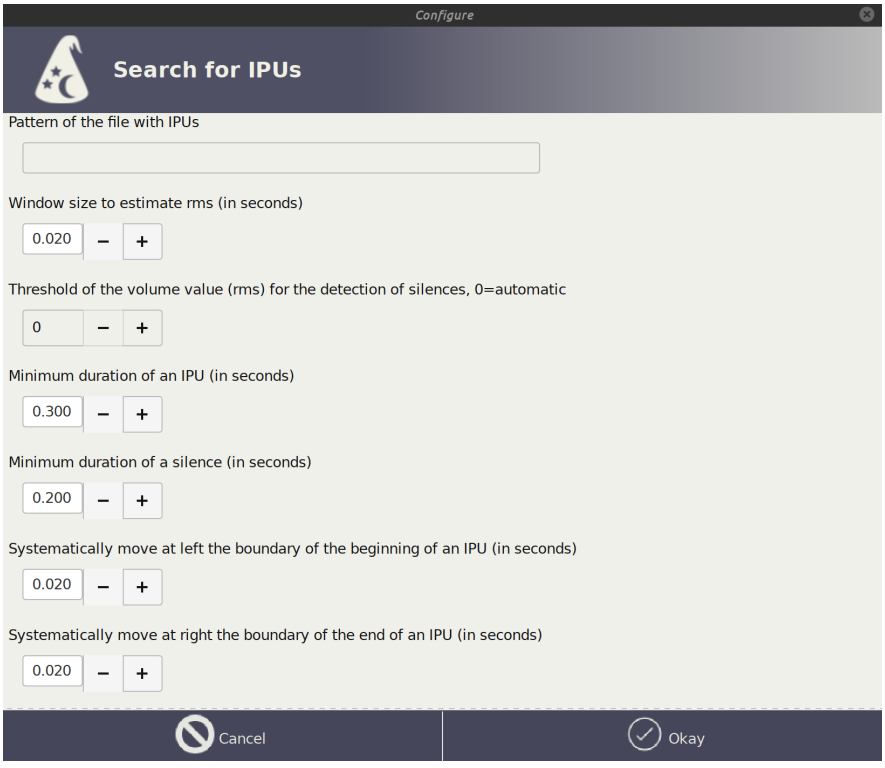

Fig. 2. Configuration with the Graphical User Interface

2.3 Discussion

If the search for IPUs algorithm is as generic as possible, some of its parameters have to be verified by the user. It was attempted to fix the default values as relevant as possible. However, most of them highly depend on the recordings. They also depend on the language and the speech-style. It is strongly recommended to the users to check these values: special care and attention should be given to each of them.

Another issue that has to be addressed in this paper concerns the fact that the algorithm removes silence intervals first then sounding ones instead of doing it in the other way around. This choice is to be explained by the concern to identify IPUs as a priority: the problem we are facing with is to search for sounding segments between silences, but not the contrary. Removing short intensity bursts first instead of short silences results in possibly removing some sounding segments with for example a low intensity, or an isolated plosive, or the beginning of an isolated truncated word, i.e. any kind of short sounding event that we are interested in. However, removing short silences first like we do results in possibly assigning a sounding interval to a silent segment.

It has to be noticed that implementing a "Search for silences" would be very easy-and-fast but at this time none of the users of SPPAS asked for.

3 Cheese! corpus

3.1 Description

"Cheese!" is a conversational corpus recorded in 2016 at the LPL - Laboratoire Parole et Langage, Aix-en-Provence, France. The primary goal of such data was a cross-cultural comparison on speaker-hearer smiling behavior in humorous and non-humorous segments of conversations in American English and French. For this reason, "Cheese!" has been recorded in respect with the American protocol [1], as far as possible.

"Cheese!" is made of 11 face-to-face dyadic interactions, lasting around 15 min each. It has been audio and video recorded in the anechoic room of the LPL. The participants were recorded with two headset microphones (AKG-C520) connected by XLR to the RME Fireface UC, which is connected with a USB cable to a PC using Audacity software. Two cameras were placed behind each of them in such a way each participant was shown from the front. A video editing software was used to merge the two videos into a single one (Figure 3) and to embed the high quality sound of the microphones.

The 22 participants were students in Linguistics at Aix-Marseille University. The participants of each pair knew each other because they were in the same class. All were French native students, and all signed a written consent form before the recordings. None of them knew the scope of the recordings.

Two tasks were delivered to the participants: they were asked to read each other a canned joke chosen by the researchers, before conversing as freely as they wished for the rest of the interaction. Consequently, although the setting played a role on some occasions, the participants regularly forgot that they were being recorded, to the extent that sometimes they reminded each other that they were being recorded when one of the participants started talking about quite an intimate topic.

In a previous study based on 4 dialogues of Cheese! [3], we observed a larger amount of laughter compared to other corpora: 3.32% of the IPUs of the read part contain

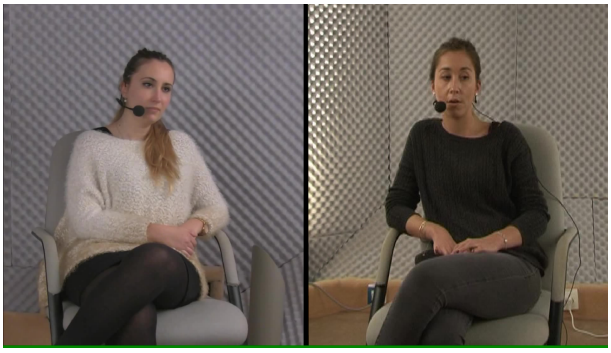

Fig. 3. Experimental design of Cheese!

laughter and 12.45% of IPUs of the conversation part. The laughter is the 5th most frequent token.

3.2 The IPUs of Cheese!

All the 11 dialogues were annotated by using the audio files re-sampled at 16,000Hz. For each of the speakers, the "Search for IPUs" automatic annotation of SPPAS was performed automatically with the following settings:

- minimum silence duration: 200 ms because it's French language;
- minimum IPUs duration: 100 ms because it's conversationnal speech;
- shift begin: 20 ms;
- shift end: 20 ms.

The IPUs were manually verified with Praat [5]. Five dialogs were verified by 2 annotators and 6 by only one.

Table 1 reports the minimum (min), mean (μ), median, σ of the statistical distribution of the rms values. The second last column indicates the resulting automatically estimated Θ. The last column indicates if the rms values were normalized. This table shows that even with the same recording conditions, the recorded rms values are ranging from very different values depending on the speaker. It confirms the need to fix a specific threshold value for each recorded file in order to get the appropriate segmentation. Fixing automatically the Θ value is then important and a great advantage for users of the software tool.

It results in the following files:

- 22 files with the IPUs SPPAS 4.1 created fully automatically;
- 22 files with the manually corrected IPUs.

4 Evaluation method

There are numerous methods and metrics to evaluate a segmentation task in the field of Computational Linguistics. Most of the methods are very useful to compare several systems and so to improve the quality of a system while developing it but their numerical result is often difficult to interpret.

Table 1. Distribution of the rms and the threshold value Θ fixed automatically

spk	min	μ	median	σ	Θ	Norm.	spk	min	μ	median	σ	Θ	Norm.
AA	12	548	58	177	295		OR	5	1009	19	160	773	
AC	6	818	371	209	510		MZ	5	1313	38	175	1056	
AW	7	595	96	164	355		CG	2	492	90	147	273	
CM	12	876	257	141	675		MCC	4	397	44	203	96	
ER	3	502	30	159	266		AG	8	328	38	168	84	
CB	4	753	63	174	495		FB	17	1058	89	194	783	
CL	3	1151	2918	15	256	x	JS	3	564	23	230	221	
LP	8	659	92	164	420		MA	3	672	123	152	446	
MA	3	608	24	178	343		PC	2	373	28	202	71	
AD	3	1680	3744	138	855	x	MD	10	844	164	199	555	
EM	4	1162	181	157	929		PR	12	427	61	188	156	

In this paper, we developped an evaluation method and a script that is distributed into the SPPAS package. It evaluates the number of manual "actions" the users had to operate in order to get the expected IPUs. We divided these manual actions into several categories described below. For a user who is going to read this paper, it will be easy to know what to expect while using this software on a conversational corpus, and to get an idea of the amount of work to do to get the expected IPUs segmentation.

In the following, the manually corrected IPUs segmentation is called "reference" and the automatic one is considered the "hypothesis". The evaluation reports the number of IPUs in the reference and in the hypothesis and the following "actions" to perform manually to transform the hypothesis into the reference:

add : number of times an IPU of the reference does not match any IPU of the hypothesis. The user had to *add* the missing IPUs;
merge : number of times an IPU of the reference matches with several IPUs of the hypothesis. The user had to *merge* two or more consecutive IPUs;
split : number of times an IPU of the hypothesis matches with several IPUs of the reference. The user had to *split* an IPU into several ones;
ignore : number of times an IPU of the hypothesis doesn't match any IPU of the reference. The user had to *ignore* a silence which was assigned to an IPU;
move_b : number of times the begin of an IPU must be adjusted;
move_e : number of times the end of an IPU must be adjusted.

The *add* action is probably the most important result to take into account. In fact, if *add* is too high it means the system failed to find some IPUs. It is critical because it means the user have to listen the whole content of the audio file to add such missing IPUs which is time consuming. If none of the IPUs is missed by the system, the user had only to listen the IPUs the system found and to check them by merging, splitting or ignoring them and by adjusting the boundaries.

In order to be exhaustive, this paper presents the *ignore* action. However, from our past experience in checking IPUs, we don't really consider this result an action to do. In practice, the user is checking IPUs at the same time of the orthographic transcription. If there's nothing interesting to transcribe, the interval is ignored: there's nothing specific to do. Moreover, we developed a plugin to SPPAS which deletes automatically these un-transcribed IPUs.

5 Results

5.1 Quantitative evaluation

We applied the evaluation method presented in the previous section on the 11 dialogs of Cheese! corpus. The evaluated actions are reported into a percentage according to the 6922 IPUs in the reference for add, merge, move_b, move_e or according to the 7343 IPUs in the hypothesis for split and ignore:

- **add:** 54 (0.79% of the IPUs in the reference)
- **merge:** 104 (1.51% of the IPUs in the reference)
- **split:** 273 (3.72% of the IPUs in the hypothesis)
- **ignore:** 724 (9.86% of the IPUs in the hypothesis are false positives)
- **move_b:** 497 (7.23% of the IPUs in reference)
- **move_e:** 788 (11.46% of the IPUs in reference)

Reducing the number of missed IPUs was one of the main objective while developping the algorithm and we can see in the evaluation results that the number of IPUs to *add* is very small: it represents only 0.78% of the IPUs of the reference. It means that the user can be confident with the tool: the sounding segments are found.

The same holds true for the *merge* action: only very few IPUs are concerned which is very good because it's relatively time-consuming to do it manually. However, the number of IPUs to *split* is relatively high which is also relatively time-consuming to do manually.

Finally, the highest number of actions to perform is to move boundaries of the IPUs but this action is done very easyly and fastly with Praat.

5.2 Qualitative evaluation

We observed the durations of the added and ignored IPUs. It is interesting to mention that the duration of the IPUs to *add* and the IPUs to *ignore* are less than the average. Actually, the duration of the IPUs of the reference is 1.46 seconds in average but the 39 IPUs we added are only 0.93 seconds in average. This difference is even more important for the IPUs we ignored: their duration is 0.315 seconds in average.

Another interesting aspect is related to the speech style of the corpus: 14.11% of the IPUs contain a laughter or a sequence of speech while laughing. These events have a major consequence on the results of the system. Most of the actions to do contain a high proportion of IPUs with a laughter or a laughing sequence:

- 11 of the 39 IPUs to *add* (28.21%);
- 86 of the 171 IPUs to *merge* (50.29%);
- 5 of the 7 IPUs to *split* (71.42%).

This analysis clearly indicates that laugher, or laughing while speaking, is responsible for a lot of the errors of the system, particularly for the actions to *split* and to *merge*. Figure 4 illustrates this problem: the first tier is the manually corrected one - the reference, and the second tier is the system output - the hypothesis.

In this scope of analyzing the errors, we also used the filtering system of SPPAS [4] that allowed us to create various tiers with: (1) the first phoneme of each IPU of the reference; (2) the first phoneme of each IPU of the reference for which a "move begin"

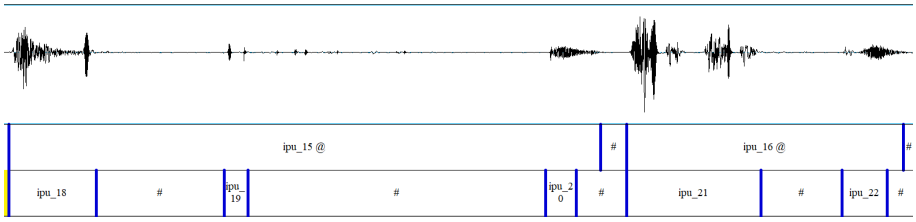

Fig. 4. Example of merged IPUs: laughter items are often problematic

action was performed; (3) the last phoneme of each IPU of the reference; (2) the last phoneme of each IPU of the reference for which a "move end" action was performed. We observed a high proportion of the fricatives /s/, /S/, /Z/ and the plosives /t/ at the beginning of the moved beginnings. On the contrary, the phonemes /w/ and /m/ which are the 2 most frequent ones in the reference are relatively less frequent in the "move begin" action. The following percentages indicate the proportion of the most frequent phonemes in the IPUs requiring the *move_b* action:

- /s/ is starting 7.96% of the IPUs of the reference but it concerns 20.08% of the IPUs of the *move_b* errors;
- /t/ is starting 4.31% IPUs of the reference but 7.19% of the *move_b* ones;
- /m/ is starting 8.66% IPUs of the reference but 5.92% of the *move_b* ones;
- /w/ is starting 11.76% IPUs of the reference but 5.70% of the *move_b* ones;
- /S/ is starting 2.83% IPUs of the reference but 4.65% of the *move_b* ones;
- /Z/ is starting 2.72% IPUs of the reference but 3.81% of the *move_b* ones;

Moreover, we observed that 10.6% of the *move_b* actions concern a laughter item.

We finally have done the same analysis for the last phoneme of the IPUs of the reference versus the last phoneme of IPUs with the *move_e* actions:

- /s/ is ending 2.67% IPUs in the reference but 7.11% in the *move_e* ones;
- /E/ is ending 9.08% IPUs in the reference but 6.85% in the *move_e* ones;
- /a/ is ending 10.42% IPUs in the reference but 6.72% in the *move_e* ones;
- /R/ is ending 5.72% IPUs in the reference but 6.32% in the *move_e* ones;
- /t/ is ending 3.88% IPUs in the reference but 6.19% in the *move_e* ones;

Like for the beginning, the /s/ and /t/ are relatively more frequent in the "move end" action than in the reference. And we observed that 17.26% of the *move_e* actions concern a laughter item which makes it the most frequently required "move end" action; but it's also the most frequent one to end an IPU with 11.11% in the reference.

Figure 5 illustrates the two actions *move_b* and *move_e* on the same IPU even if this situation is quite rare. In this example, the first phoneme is /s/ and the last one is /k/.

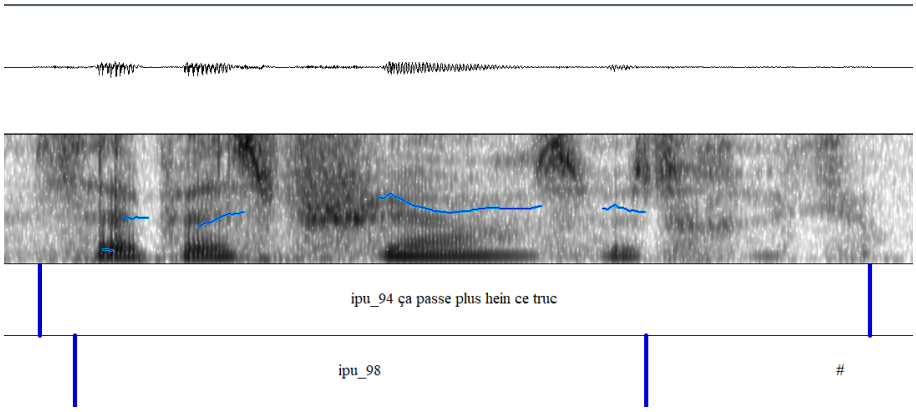

Fig. 5. Example of the *move_b* and *move_e* actions on an IPU with a low rms

6 Conclusion

This paper described a method to search for inter-pausal units. This program is part of SPPAS software tool. The program has been evaluated on the 11 dialogues of about 15 minutes each of Cheese! corpus, a corpus made of both read speech (1 minute) and spontaneous speech.

We observed that the program allowed to find properly the IPUs, even on this particularly difficult corpus of conversations. To check the output of this automatic system, we had to perform the following actions on the IPUs the system found: to add new ones, to merge, to split, to ignore; and to perform the following actions on their boundaries: to move the beginning, to move the end. The analysis of the results showed that laughter are responsible for a large share of the errors. This is mainly because a laughter is a linguistic unit but acoustically it's often an outcome of alternate sounding and silence segments (Figure 4).

7 Acknowledgments

We address special thanks to the Centre d'Expérimentation de la Parole (CEP), the shared experimental platform for the collection and analysis of data, at LPL.

References

1. Attardo, S., Pickering, L., Baker, A.: Prosodic and multimodal markers of humor in conversation. Pragmatics & Cognition 19(2), 224–247 (2011)
2. Bigi, B.: SPPAS - Multi-lingual Approaches to the Automatic Annotation of Speech. The Phonetician 111–112, 54–69 (2015)
3. Bigi, B., Meunier, C.: Automatic segmentation of spontaneous speech. Revista de Estudos da Linguagem. International Thematic Issue: Speech Segmentation 26(4) (2018)

4. Bigi, B., Saubesty, J.: Searching and retrieving multi-levels annotated data. In: Proceedings of Gesture and Speech in Interactioni - 4th edition. pp. 31–36. Nantes, France (2015)
5. Boersma, P., Weenink, D.: Praat: Doing phonetics by computer. [Computer Software] Amsterdam: Department of Language and Literature, University of Amsterdam. . http://www.praat.org/ (2011)
6. Leech, G.: Introducing corpus annotation. Corpus Annotation: Linguistic Information from Computer Text Corpora. Longman, London pp. 1–18 (1997)
7. Priego-Valverde, B., Bigi, B., Attardo, S., Pickering, L., Gironzetti, E.: Is smiling during humor so obvious? A cross-cultural comparison of smiling behavior in humorous sequences in american english and french interactions. Intercultural Pragmatics 15(4), 563–591 (2018)
8. Priego-Valverde, B., Bigi, B., Amoyal, M.: "Cheese!": a Corpus of Face-to-face French Interactions. A Case Study for Analyzing Smiling and Conversational Humor. In: Proceedings of The 12th Language Resources and Evaluation Conference. pp. 467–475. European Language Resources Association, Marseille, France (2020)
9. Wallis, S., Nelson, G.: Knowledge discovery in grammatically analysed corpora. Data Mining and Knowledge Discovery 5, 307–340 (2001)