Search for Inter-Pausal Units: application to Cheese! corpus

Brigitte Bigi, Béatrice Priego-Valverde

Laboratoire Parole et Langage, 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 - 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 the use and the evaluation of this feature on Cheese! corpus, a corpus of read then conversational speech between two participants. The paper reports the number of manual actions that are required for a user to check the automatic annotation: add new IPUs, ignore un-relevant ones, move boundaries, etc. Such evaluation validates the proposed method.

1. Introduction

In recent years, the SPPAS software tool (Bigi, 2015) has been developed by the first author to automatically produce annotations and to analyze annotated data. SPPAS 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, it allows to perform speech segmentation of a recorded speech audio and its orthographic transcription (Bigi and Meunier, 2018). In order to prepare the latter, an automatic search for Inter-Pausal Units (IPUs) is also proposed. The orthographic transcription is performed manually inside the IPUs the system found (Figure 1).

Figure 1: Transcription process when using SPPAS

This paper presents the automatic annotation "Search for IPUs" included into SPPAS. Given a speech recording, the goal of this task is to generate an annotation file in which the sounding segments between silences are marked. This automatic annotation is applied and evaluated on the conversational French corpus 'Cheese!' (Priego-Valverde et al., 2018).

2. The method to search for IPUs

2.1. Algorithm and settings

At a first stage, the Root-Mean-Square (rms) values are estimated on windows of a fixed duration of the audio recording. The duration of such windows is fixed by default to 20 ms and can be configured by the user.

The statistical distribution of such rms values is analyzed to fix automatically a threshold value Θ. The latter is used to decide if each window is either a "silence" - rms is under the threshold, or a "sounding" one - rms is higher than the threshold. The value of Θ is fixed as follow:

$$\Theta = min + \mu - \delta$$

δ is generally fixed to 1.5σ where σ is the coefficient of variation. 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. Actually, if the audio is not as good as expected, outliers values are replaced by the mean and the analysis is performed on the new normalized distribution.

It has to be noticed that the threshold value strongly depends on the quality of the recording and the value fixed automatically may not be appropriate on some recordings. Consequently, the user can fix it manually.

Each window is then evaluated and the intervals below and above the threshold are identified respectively as silence and sounding. The neighboring silent and neighboring sounding windows are grouped into intervals. The resulting silent intervals with a too small duration are removed. This minimum duration is fixed to 200 ms by default which is often relevant for French, however it should be changed to 250 ms for English language. This difference is mainly due to the voiceless velar plosive /k/ in which the silence before the plosion could be longest than the duration fixed by default.

The next step of the algorithm starts by re-grouping neighboring sounding intervals that resulted because of the removal of the too short silences. The resulting sounding intervals with a too small duration are then removed. This minimum duration is fixed to 300 ms by default. This value have 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. However, in spontaneous speech like conversation, it has to be lowered mainly because of some isolated feedback like 'mh' or 'ah' which could be missed by the system.

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.

This algorithm and its settings can be summarized as follow:

1. fix a window length to estimate rms (default is 20 ms);

2. estimate rms value 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).

2.2. Optional settings

From our past experience of distributing this tool, we received feedback of users. They allowed to improve the values to be fixed by default mentioned in the previous section. They 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 any sounding 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`.

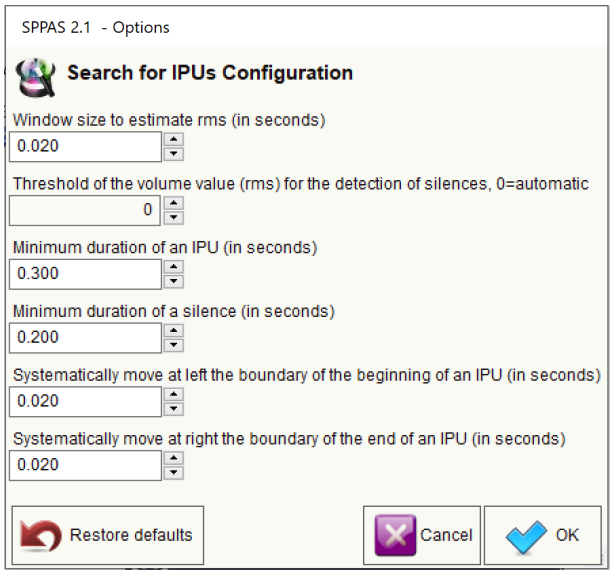

Figure 2: Configuration with the Graphical User Interface

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, in particular they depend on the language, the speech-style, and the recording condition. 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 can 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 it to search for sounding segments between silences, 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 don't want to miss. It's clearly not what we are expecting when we are searching for IPUs. However, removing short silences first like it's done, 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 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 (Attardo et al., 2011), as far as possible.

Cheese! is composed 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.

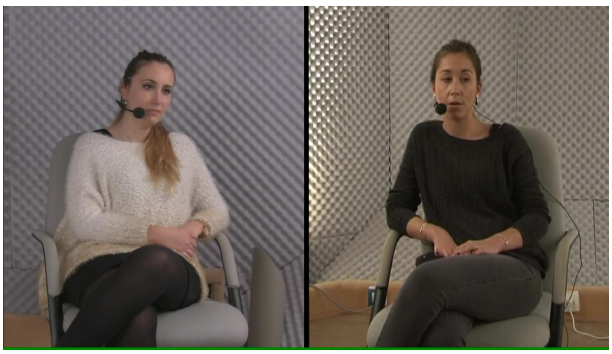

Figure 3: Experimental design of Cheese!

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.

It has to be noticed that in a previous study based on 4 dialogues of Cheese! (Bigi and Meunier, 2018), it was observed a larger amount of laughter compared to other corpora: 3.32% of the IPUs of the read part contain laughter and 12.45% of IPUs of the conversation part. The laughter is the 5th most frequent token.

At the time of writing this paper, 5 dialogues were annotated. For each of the speakers, the "Search for IPUs" automatic annotation of SPPAS was performed automatically. Table 1 reports the minimum (min), mean (μ), median, σ and the resulting threshold Θ. The last column indicates if the rms values were normalized. These IPUs were manually verified by the authors, with Praat (Boersma and Weenink, 2018). It results in 10 files with the IPUs automatically found and the corresponding 10 files with the expected IPUs.

spk	min	μ	median	σ	Θ	Norm.
AD	3	1682	3749	138	859	x
AG	6	842	39	158	611	
CL	3	1156	2913	157	269	x
CM	12	878	264	140	679	
ER	15	659	77	168	422	
JS	7	1130	47	230	791	
MA	6	1015	313	178	754	
MCC	5	399	44	202	151	
MD	10	848	164	198	561	
PC	4	624	45	201	325	

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

4. Evaluation metric

There are numerous methods and metrics to evaluate a segmentation task in the field of Computational Linguistics. Most of such 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 difficult to interpret.

In the scope of writing this paper, we preferred to evaluate the number of manual "actions" the users will have to do in order to get the expected IPUs. We divided these manual actions to operate into several categories described in details 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 a correct 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 IPUs of the reference that do not match any IPU of the hypothesis. The user has to *add* the missing IPUs;

merge : number of time an IPU of the reference matches with several IPUs of the hypothesis. The user has to *merge* two or more consecutive IPUs;

split : number of time an IPU of the hypothesis matches with several IPUs of the reference. The user has to *split* an IPU into several ones;

ignore : number of IPUs of the hypothesis that don't match any IPU of the reference. The user has 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.

All these actions are reported into a percentage according to the number of IPUs in the reference (add, merge, move_b, move_e) or according to the number of IPUs in the hypothesis (split, ignore).

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's important because it means the user will 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 will have 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, in this paper we present the *ignore* action. However, from our past experience in checking IPUs, we don't really consider this result an action to do. To save time, in practice, we are checking IPUs at the same time we are transcribing speech. If there's nothing interesting to transcribe, we just ignore the interval. There's no really a specific action to do here except for those who would like to delete them. And for this purpose, we developed a plugin to SPPAS to delete automatically un-transcribed IPUs.

5. Results

Table 2 presents the evaluation results of the "Search for IPUs" on Cheese! corpus. The annotation was performed with the following parameters:

- minimum silence duration: 200 ms
- minimum IPU duration: 100 ms
- shift begin: 20 ms
- shift end: 20 ms

The columns 'ref' and 'hyp' indicate the number of IPUs respectively in the reference - i.e. after the manual check, and in the hypothesis - i.e. the result of the automatic system.

Reducing the number of missed IPUs was one of the objective while developing the algorithm and we can see in the table that the number of IPUs to *add* is very small: it represents only 1.21% of the IPUs of the reference. The same holds true for the *split* action: only very few IPUs are concerned. On the other hand, the number of IPUs to *merge* is relatively high.

It is interesting to mention that 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%).

speaker	ref	hyp	*add*	*merge*	*split*	*ignore*	*move_b*	*move_e*
AD	512	509	5	8	4	1	36	82
AG	278	304	1	20	0	6	23	53
CL	349	363	3	18	2	3	35	42
CM	319	348	5	26	0	6	49	53
ER	224	239	1	7	0	7	19	27
JS	327	370	9	36	0	12	31	63
MA	293	318	0	15	0	11	32	55
MCC	233	271	2	7	1	34	15	22
MD	368	410	3	22	0	18	33	67
PC	324	327	10	12	3	1	41	35
total	3227	3459	39	171	7	99	314	499
			1.21%	5.30%	0.66%	2.86%	9.73%	15.46%

Table 2: Results of the search for IPUs

This analysis clearly indicates that the laugh, 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.

Finally, the highest number of actions to perform is to move boundaries of the proposed IPUs. We analyzed the first phoneme of the IPUs with *move_b* action and without surprise we observed a high proportion of the fricatives /s/, /S/ and /Z/ and the voiceless plosives /t/ and /k/. The following percentages indicate the proportion of such phonemes in both the IPUs of the reference and the IPUs requiring the *move_b* action:

- /s/ is starting 7.55% of the IPUs of the reference but it concerns 22.26% of the IPUs of the *move_b* errors;
- /S/ is starting 2.42% IPUs of the reference but 8.84% of the *move_b* ones;
- /Z/ is starting 3.04% IPUs of the reference but 5.79% of the *move_b* ones;
- /t/ is starting 4.97% IPUs of the reference but 9.14% of the *move_b* ones;
- /k/ is starting 3.39% IPUs of the reference but 5.79% of the *move_b* ones.

Moreover, we observed that 13.7% of the *move_b* actions concern a laughter item.

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

- /t/ is ending 4.01% IPUs in the reference but 10.96% in the *move_e* ones;
- /s/ is ending 2.42% IPUs in the reference but 9.16% in the *move_e* ones;

And we also observed that 22.5% of of the *move_e* actions concern a laughter item.

Figure 5 illustrates the two actions *move_b* and *move_e* on the same IPU even if this situation is quite rare: only 77 IPUs require both actions. In this example, the first phoneme is /s/ and the last one is /k/.

6. Conclusion

This paper described a method to search for IPUs. This program is part of SPPAS software tool. The program has been evaluated on Cheese! corpus, a corpus made of both read speech and spontaneous speech. Five dialogues of about 15 minutes each were used.

We observed that if the parameters are fixed properly, the program allows to find properly the IPUs, even on a 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 (1.2%), to merge (5.3%), to split (0.7%), to ignore (2.9%); and to perform the following actions on their boundaries: to move the beginning (9.7%), to move the end (15.5%). 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.

8. References

Attardo, Salvatore, Lucy Pickering, and Amanda Baker, 2011. Prosodic and multimodal markers of humor in conversation. *Pragmatics & Cognition*, 19(2):224–247.

Bigi, Brigitte, 2015. SPPAS - Multi-lingual Approaches to the Automatic Annotation of Speech. *The Phonetician*, 111–112:54–69.

Bigi, Brigitte and Christine Meunier, 2018. Automatic segmentation of spontaneous speech. *Revista de Estudos da Linguagem. International Thematic Issue: Speech Segmentation*, 26(4).

Boersma, Paul and David Weenink, 2018. Praat: doing phonetics by computer [computer program], version 6.0.37, retrieved 14 march 2018 from http://www.praat.org/.

Priego-Valverde, Béatrice, Brigitte Bigi, Salvatore Attardo, Lucy Pickering, and Elisa Gironzetti, 2018. 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.

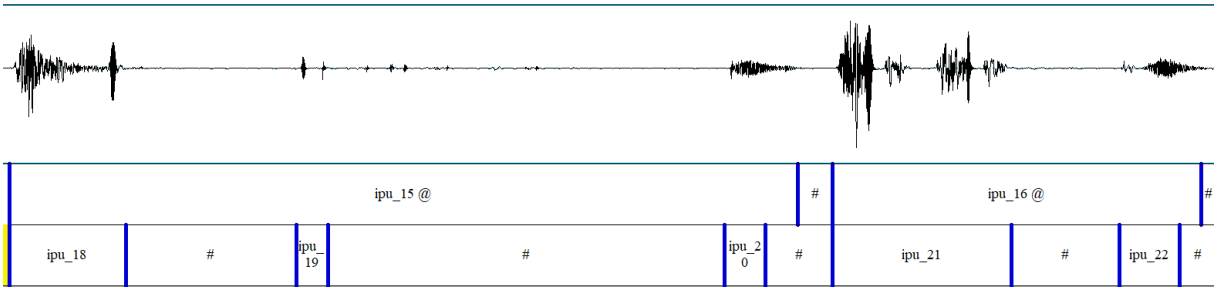

Figure 4: Example of merged IPUs: laughter items are often problematic

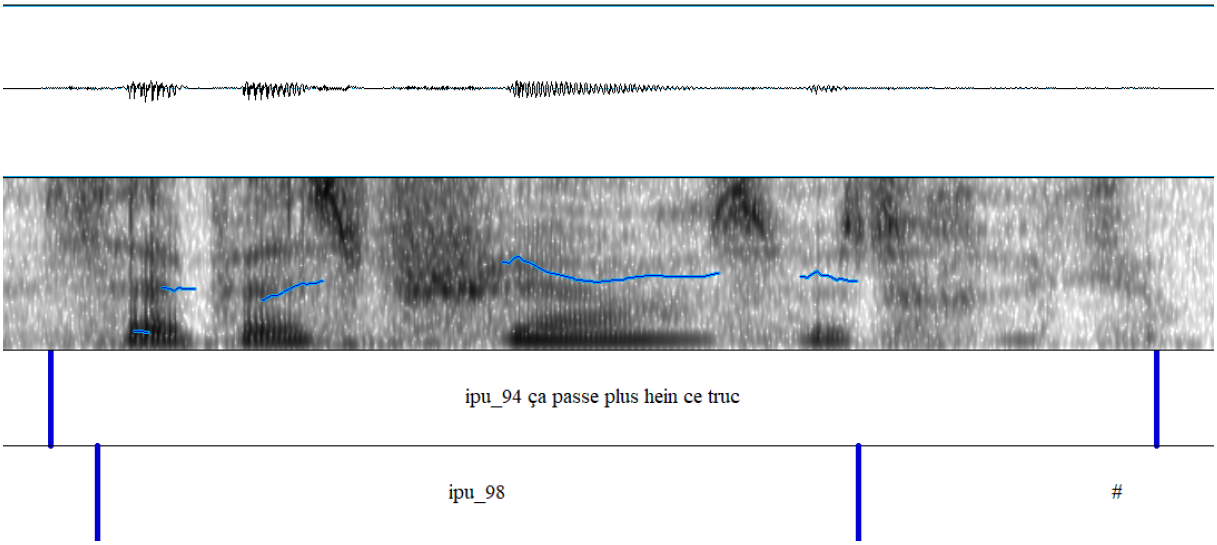

Figure 5: Example of the *move_b* and *move_e* actions