What's new in SPPAS 1.5?

Brigitte Bigi[1], *Daniel Hirst*[1,2]

[1]LPL, CNRS, Aix-Marseille Université, Aix-en-Provence, France
[2]School of Foreign Languages, Tongji University, Shanghai, China
brigitte.bigi@lpl-aix.fr, daniel.hirst@lpl-aix.fr

Abstract

During Speech Prosody 2012, we presented *SPPAS*, SPeech Phonetization Alignment and Syllabification, a tool to automatically produce annotations which include utterance, word, syllabic and phonemic segmentations from a recorded speech sound and its transcription. SPPAS is open source software issued under the GNU Public License. SPPAS is multi-platform (Linux, MacOS and Windows) and it is specifically designed to be used directly by linguists in conjunction with other tools for the automatic analysis of speech prosody. This paper presents various improvements implemented since the previously described version.

Index Terms: phonetic, annotation, segmentation, intonation

1. Introduction

During Speech Prosody 2012, we presented version 1.3 of SPPAS (SPeech Phonetization Alignment and Syllabification). SPPAS was presented as a tool to produce automatic annotations which include utterance, word, syllabic and phonemic segmentations from a recorded speech sound and its transcription. The resulting alignments are a set of TextGrid files, the native file format of the Praat software [1] which has become the most popular tool for phoneticians today. SPPAS generates separate TextGrid files for 1/ utterance segmentation, 2/ word segmentation, 3/ syllable segmentation and 4/ phoneme segmentation.

An important point for a software which is intended to be widely distributed is its licensing conditions. SPPAS uses only resources and tools which can be distributed under the terms of the GNU Public License. SPPAS tools and resources are currently available at the URL:

http://www.lpl-aix.fr/∼bigi/sppas/

Since the version presented in [2], we continued to improve the tool. Our improvements are related to the 4 following aspects:

1. Technical stuff: multi-platform, easy to install, UTF-8 support;
2. Graphical User Interface: improved ergonomics, documentation and help, some components added;
3. Annotations: Momel and INTSINT added; Tokenization added; IPU-segmentation improved;
4. Resources: acoustic model for Chinese changed, Taiwanese support, conversion to SAMPA.

The new SPPAS architecture can be summarized as:

- a set of automatic annotation tools,
- a set of components,
- two solutions to use them:
  1. a Graphical User Interface (GUI) to use SPPAS which is as ”user-friendly” as possible;
  2. a set of tools, each one essentially independent of the others, that can be run on its own at the level of the shell.

2. Technical stuff

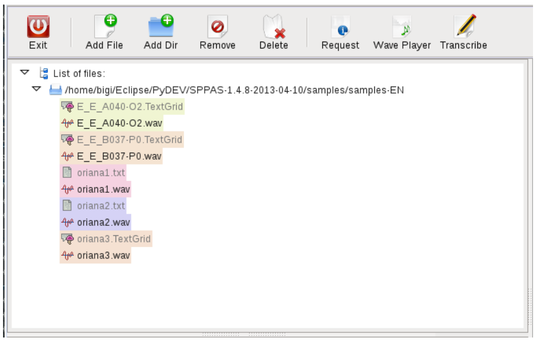
Since version 1.4, SPPAS is implemented with the programming language *python*. This allows the tool to work under Linux, Mac-OSX and Windows®. It is also much easier to install.

In the previous version, only TextGrid files were supported. The current version can import files from Transcriber [3] and Elan [4] softwares. We also fixed the encoding to UTF-8 only.

3. Graphical User Interface

The GUI consists of two main area, named the file list panel (FLP) and the automatic annotation panel (AAP).

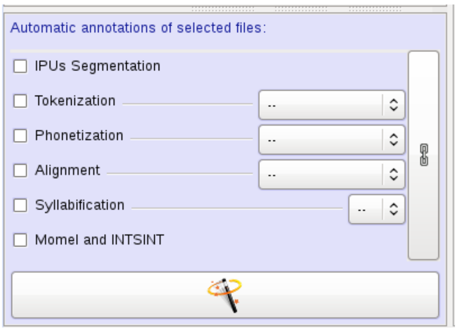
The FLP displays a set of buttons and a tree-style list. The list contains Directories and Files which the user has added, but only files that SPPAS can handle (recognised by the file extension). The FLP makes it possible to exit the tool and to manage the list: add files, add directories, remove, delete, export.



List of files:
/home/bigi/Eclipse/PyDEV/SPPAS-1.4.8-2013-04-10/samples/samples-EN
E_E_A040-O2.TextGrid
E_E_A040-O2.wav
E_E_B037-P0.TextGrid
E_E_B037-P0.wav
oriana1.txt
oriana1.wav
oriana2.txt
oriana2.wav
oriana3.TextGrid
oriana3.wav

Figure 1: *The file list panel.*

The AAP consists of a list of buttons to check, the annotation name and buttons to fix the language of each annotation. A specific language can be selected for each annotation depending on the resources available in the package. This allows the users to add their own resources or to copy/modify existing resources.



Automatic annotations of selected files:

□ IPUs Segmentation

□ Tokenization --

□ Phonetization --

□ Alignment --

□ Syllabification --

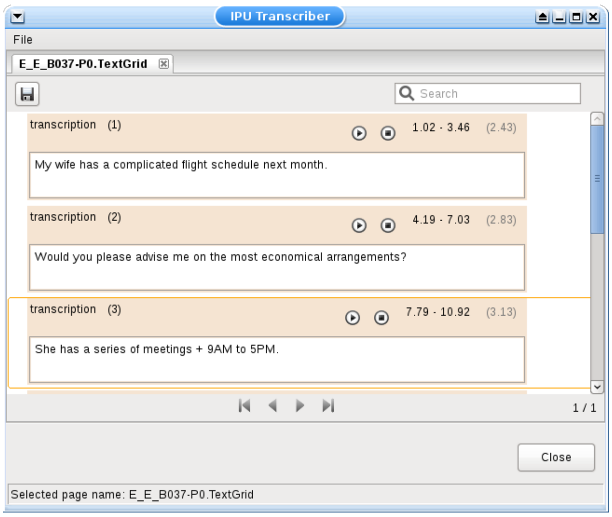
□ Momel and INTSINT

Figure 2: *The automatic annotation panel.*

As SPPAS is designed to be used directly by linguists, another important improvement is related to the Help and the Documentation. We paid particular attention to this. Finally, to facilitate the use of our tool, we decided to add some extra components. Currently, three components are available: 1/ **wav player** is a simple tool used to play sounds; 2/ **transcribe** is a tool dedicated to speech transcription; 3/ **requests** is a set of functionalities related to the annotation manipulation.

3.1. Transcribe

The key-point of this component is that it automatically performs a speech/silence segmentation. Then, only speech segments are displayed (see Figure 3). If more than one sound file has to be transcribed (as for a dialogue for example), speech segments are displayed interlaced to facilitate the transcription process.



transcription (1) 1.02 - 3.46 (2.43)

My wife has a complicated flight schedule next month.

transcription (2) 4.19 - 7.03 (2.83)

Would you please advise me on the most economical arrangements?

transcription (3) 7.79 - 10.92 (3.13)

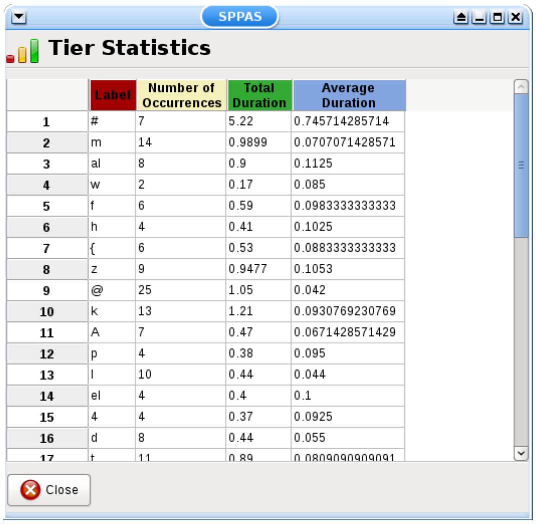
She has a series of meetings + 9AM to 5PM.

Selected page name: E_E_B037-P0.TextGrid

Figure 3: *The transcription frame.*

3.2. Requests

We added a component to get information, modify and request annotated files (see Figure 4). This allows the user to manage annotated files and the tiers of these files: rename, delete, cut, copy, paste duplicate, move up, move down, view. We also added a frame that prints elementary statistics (as in Figure 5).



Tier Statistics

	Label	Number of Occurrences	Total Duration	Average Duration
1	#	7	5.22	0.745714285714
2	m	14	0.9899	0.0707071428571
3	al	8	0.9	0.1125
4	w	2	0.17	0.085
5	f	6	0.59	0.0983333333333
6	h	4	0.41	0.1025
7	{	6	0.53	0.0883333333333
8	z	9	0.9477	0.1053
9	@	25	1.05	0.042
10	k	13	1.21	0.0930769230769
11	A	7	0.47	0.0671428571429
12	p	4	0.38	0.095
13	l	10	0.44	0.044
14	el	4	0.4	0.1
15	4	4	0.37	0.0925
16	d	8	0.44	0.055
17	t	11	0.89	0.0809090909091

Figure 5: *The statistics of a tier.*

Finally, we added an advanced filtering tool. In the following, X represents an interval, $L(X)$ the label of X, and $L(.)$ one label to find. We thus propose to select intervals depending on their label with the following capabilities:

- $L(X) = L(.)$, exact match: the labels must strictly correspond,
- $L(X) \in L(.)$, contains: the label of the tier contains the given label,
- $L(X) \sqsubset L(.)$, starts with: the label of the tier starts with the given label,
- $L(X) \sqsupset L(.)$, ends with: the label of the tier ends with the given label.

All these matches can be used in their negative form. To cope with specific needs, a multiple pattern selection has been implemented to search n patterns $L_1(.), L_2(.), \cdots, L_n(.)$ as:

$$X : [L(X)\ op\ (L_1(.) \vee L_2(.) \vee \cdots \vee L_n(.))]$$

where *op* represents one of the relations $=, \in, \sqsubset, \sqsupset$. At last, the proposed filtering system makes it possible to fix duration constraints. Let $D_m(.)$ be a minimal duration, $D_M(.)$ be a maximal duration and $D(X)$ be the duration of interval X. Duration constraints are written as:

- $X : [D(X) > D_m(.)]$, to fix a minimal duration on X,
- $X : [D(X) < D_M(.)]$, to fix a maximal duration on X.

For example, the request "Extract all words starting by "ch" with a duration of at least 100ms" is expressed as:

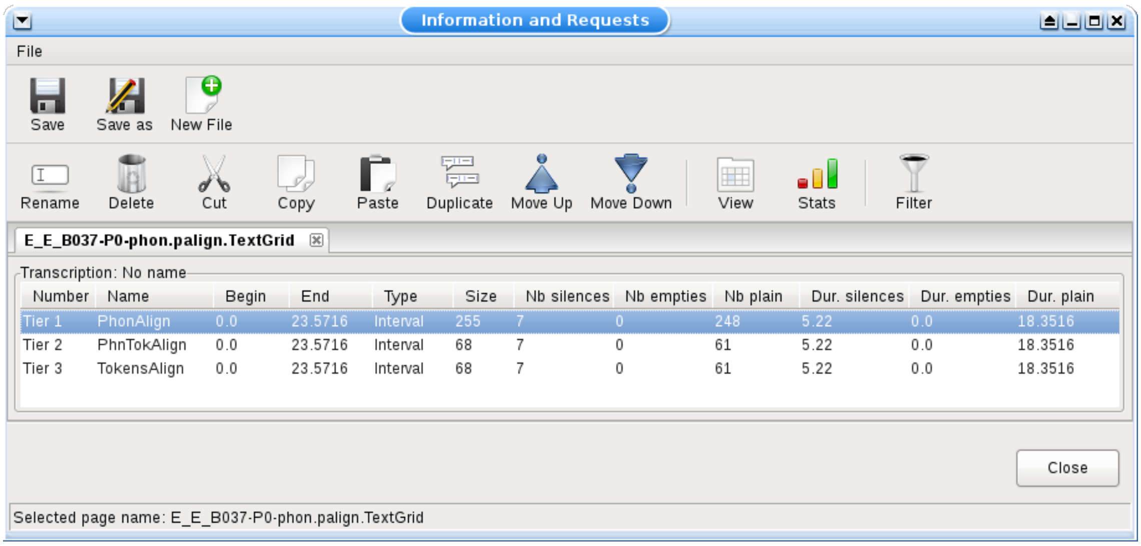
$$X : [L(X) \sqsubset L(ch)]\ \{eq\}\ [D(X) > D_m(100ms)]$$

These constraints can be applied to a whole tier or to just a part of the tier by fixing a start time and an end time.

4. Annotations

4.1. IPU segmentation

Inter-Pausal Units (IPUs) segmentation consists in aligning the macro-units of a document (based on their transcription) with



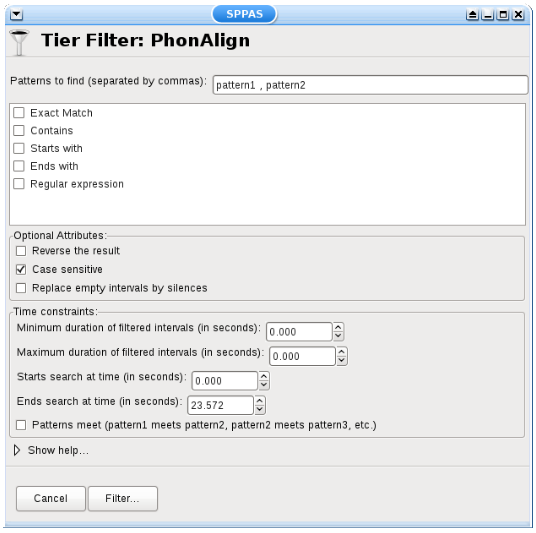
E_E_B037-P0-phon.palign.TextGrid

Transcription: No name

Number	Name	Begin	End	Type	Size	Nb silences	Nb empties	Nb plain	Dur. silences	Dur. empties	Dur. plain
Tier 1	PhonAlign	0.0	23.5716	Interval	255	7	0	248	5.22	0.0	18.3516
Tier 2	PhnTokAlign	0.0	23.5716	Interval	68	7	0	61	5.22	0.0	18.3516
Tier 3	TokensAlign	0.0	23.5716	Interval	68	7	0	61	5.22	0.0	18.3516

Selected page name: E_E_B037-P0-phon.palign.TextGrid

Figure 4: *The frame to manipulate annotated files.*



Tier Filter: PhonAlign

Patterns to find (separated by commas): pattern1 , pattern2

□ Exact Match
□ Contains
□ Starts with
□ Ends with
□ Regular expression

Optional Attributes:
□ Reverse the result
☑ Case sensitive
□ Replace empty intervals by silences

Time constraints:
Minimum duration of filtered intervals (in seconds): 0.000
Maximum duration of filtered intervals (in seconds): 0.000
Starts search at time (in seconds): 0.000
Ends search at time (in seconds): 23.572
□ Patterns meet (pattern1 meets pattern2, pattern2 meets pattern3, etc.)

▷ Show help...

Figure 6: *Filtering a tier.*

the corresponding sound. A recorded speech file with the .wav extension should correspond to each .txt file. The segmentation provides a TextGrid file with one tier named "IPU". IPUs Segmentation annotation performs a simple silence detection if no transcription is available (the volume is automatically adjusted). Current version allows to fix a shift value to speech boundaries.

4.2. Tokenization

Tokenization is the process of segmenting a text into tokens. In principle, any system that deals with unrestricted text needs the text to be normalised. Texts contain a variety of "non-standard" token types such as digit sequences, words, acronyms and letter sequences in all capitals, mixed case words, abbreviations, roman numerals, URL's and e-mail addresses... Normalising or rewriting such texts using ordinary words is then an important issue.

SPPAS implements a generic approach for text normalisation, in view of developping a multi-purpose multi-lingual text corpus. This approach consists in splitting the text normalisation problem into a set of minor sub-problems each of which is as language-independent as possible. This approach is described in [5].

The Tokenization process takes as input a transcription that can be enriched by various phenomena, such as:

- truncated words, noted as a '-' at the end of the token string (an ex- example);
- liaisons, noted between '=' (an =n= example);
- noises, noted by a '*' (only for French and Italian);
- laughs, noted by a '@' (only for French);
- short pauses, noted by a '+' (a + example);
- elisions, mentioned in parenthesis;
- specific pronunciations with brackets [example,eczap];
- comments with braces or brackets {this} or [this];
- morphological variants with <like,lie ok>,
- proper name annotation, like $John Doe$.

4.3. Phonetisation

Phonetisation, also called grapheme-phoneme conversion, is the process of representing sounds with phonetic signs. The phonetisation is the equivalent of a sequence of dictionary look-ups. It is generally assumed that all words of the speech transcription are mentioned in the pronunciation dictionary. Otherwise, SPPAS implements a language-independent algorithm to phonetise unknown words. At this stage, it consists in exploring the unknown word from left to right and then finding the longest strings in the dictionary. Since this algorithm uses the dictionary, the quality of such a phonetisation will depend on this resource.

4.4. Alignment

Phonetic alignment consists in a time-matching between a given speech utterance and a phonetic representation of the utterance. For each utterance, the orthographic and phonetic transcriptions are used. SPPAS performs an alignment to identify the temporal boundaries of phones and words. 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. The quality of such an alignment will depend on this resource.

4.5. Syllabification

The syllabification of phonemes is performed with a rule-based system previously described for French in [6]. A new set of rules was developed to deal with Italian.

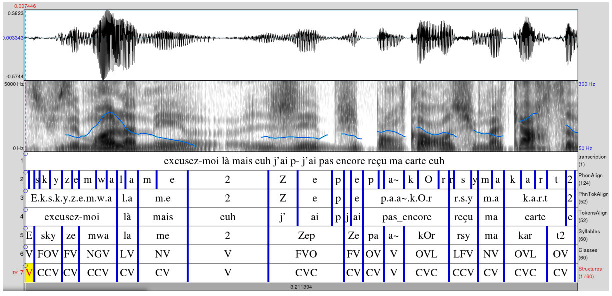

0.3823
0.003343
-0.5744
5000 Hz
0 Hz
300 Hz
50 Hz
1 excusez-moi là mais euh j'ai p- j'ai pas encore reçu ma carte euh transcription (1)
2 s k y z e m w a l a m e 2 Z e p e p a~ k O r r s y m a k a r t 2 PhonAlign (124)
3 E.k.s.k.y.z.e.m.w.a l.a m.e 2 Z e p e p.a.a~.k.O.r r.s.y m.a k.a.r.t 2 PhnTokAlign (52)
4 excusez-moi là mais euh j' ai p j ai pas_encore reçu ma carte e TokensAlign (52)
5 E sky ze mwa la me 2 Zep Ze pa a~ kOr rsy ma kar t2 Syllables (60)
6 V FOV FV NGV LV NV V FVO FV OV V OVL LFV NV OVL OV Classes (60)
7 V CCV CV CCV CV CV V CVC CV CV V CVC CCV CV CVC CV Structures (1/60)
3.211394

Figure 7: *SPPAS output example on French spontaneous speech.*

4.6. Momel and INTSINT

Momel (modelling melody) [7, 8] is an algorithm for the automatic modelling of fundamental frequency (F0) curves using a technique called assymmetric modal quadratic regression. This technique makes it possible by an appropriate choice of parameters to factor an F0 curve into two components:

1. a macroprosodic component represented by a a quadratic spline function defined by a sequence of target points <ms, hz>.
2. a microprosodic component represented by the ratio of each point on the F0 curve to the corresponding point on the quadratic spline function.

Since several different techniques of F0 extraction are possible, Momel requires a file containing the F0 values detected from the signal.

Encoding of F0 target points using the "INTSINT" system [9] assumes that pitch patterns can be adequately described using a limited set of tonal symbols, T, M, B, H, S, L, U, D (standing for : Top, Mid, Bottom, Higher, Same, Lower, Upstepped, Downstepped respectively) each one of which characterises a point on the fundamental frequency curve .

The rationale behind the INTSINT system is that the F0 values of pitch targets are programmed in one of two ways: either as absolute tones T, M, B which are assumed to refer to the speaker's overall pitch range (within the current Intonation Unit), or as relative tones H, S, L, U, D assumed to refer only to the value of the preceding target point.

A distinction is made between non-iterative H, S, L and iterative U, D relative tones since in a number of descriptions it appears that iterative raising or lowering uses a smaller F0 interval than non-iterative raising or lowering.

5. Resources

Since the version we presented in [2], we continued to improve the resources as:

- Chinese: can deal with chinese characters or with pinyin, new Chinese acoustic model, and some minor changes in the dictionary;
- English acoustic models updated
- new Italian acoustic model;
- partially Taiwanese support.

A new French model is under construction. All models were converted to Sampa.

6. Conclusions

SPPAS is specifically designed with the aim of providing a tool for phoneticians rather than for computer-scientists, because no such a tool is currently available under a GPL license.

Current development is in progress to continue to improve the accessibility, to add new language, new annotations, and new components.

7. Acknowledgements

This work is supported by ORTOLANG: http://www.ortolang.fr/

8. References

[1] P. Boersma and D. Weenink, "Praat: doing phonetics by computer, http://www.praat.org," 2009.

[2] B. Bigi and D.-J. Hirst, "SPeech Phonetization Alignment and Syllabification (SPPAS): a tool for the automatic analysis of speech prosody," in *Proc. of Speech Prosody*, Tongji University Press, Ed., Shanghai (China), 2012.

[3] TranscriberAG, "A tool for segmenting, labeling and transcribing speech. [computer software] paris: Dga," http://transag.sourceforge.net/, 2011.

[4] H. Sloetjes, A. Russel, and A. Klassmann, "Elan: a free and open source multimedia annotation tool," 2010.

[5] B. Bigi, "A multilingual text normalization approach," in *2nd Less-Resourced Languages workshop, 5th Language & Technology Conference*, Poznàn (Poland), 2011.

[6] B. Bigi, C. Meunier, I. Nesterenko, and R. Bertrand, "Automatic detection of syllable boundaries in spontaneous speech," in *Language Resource and Evaluation Conference*, La Valetta (Malta), 2010, pp. 3285–3292.

[7] D.-J. Hirst and R. Espesser, "Automatic modelling of fundamental frequency using a quadratic spline function," *Travaux de l'Institut de Phonétique d'Aix*, vol. 15, pp. 75–85, 1993.

[8] D.-J. Hirst, "A Praat plugin for Momel and INTSINT with improved algorithms for modelling and coding intonation," in *Proceedings of the XVIth International Conference of Phonetic Sciences*, Saarbrucken, 2007, pp. 1233–1236.

[9] ——, "The analysis by synthesis of speech melody: from data to models." *Journal of Speech Sciences*, vol. 1, no. 1, pp. 55–83, 2011.