Natural Language Processing

Politecnico di Milano

Polo di Como

Prof. Licia Sbattella - AA 2012 - 2013

Assignment: text & speech analysis

2013-6-14 V1.0

1 Main Goal

You are required to generate a graphical representation of a dialogue between two persons. Such representation will leverage both audio and textual analyses.

2 The process to follow

2.1 Getting the dialogue

Find and download a spoken dialogue, involving two speakers, possibly with high levels of emotive speaking. Save the audio file as *dialogue.wav* [1]

Prepare a transcript of the dialogue; something like:

Alice: how are you?

Bob: fine thanks

Alice: tell me about your project

Do not transcribe non-words, like "hem", "mmm", cough, etc.

Save the result as *dialogue.txt*

Prepare also a "flat" version of the same file, removing speaker names and punctuation:

How are you fine thanks tell me about your project

Save this file as *dialogue-flat.txt*

2.2 Text-to-speech alignment & tagging

Using SPPAS, try to generate an alignment between the speech (*dialogue.wav*) and the transcription (*dialogue-flat.txt*).

[1] You can use tools like Audacity to create a WAV file, starting from an MP3.

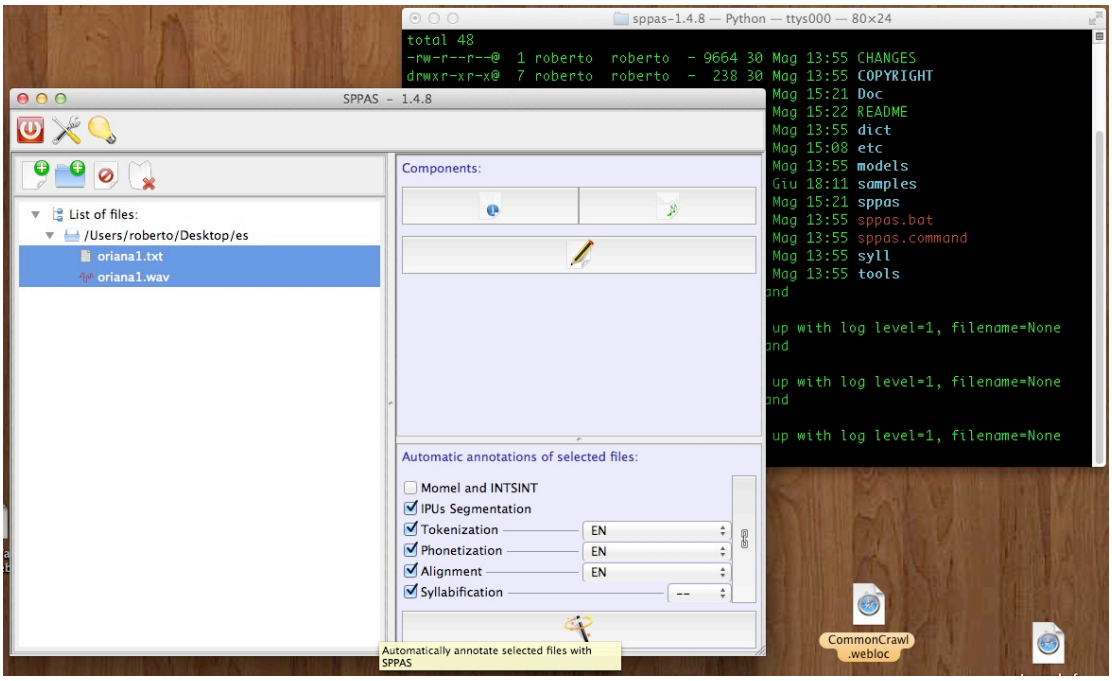

Figure 1 - SPPAS aligning the file "oriana1.wav" and "oriana1.txt"

The result will be a TextGrid file (e.g., *oriana1-phon.palign.TextGrid*). Use Praat to edit such a file (the tier we need is the one named “TokensAlign”). You can remove the other tiers, just to clean a bit...

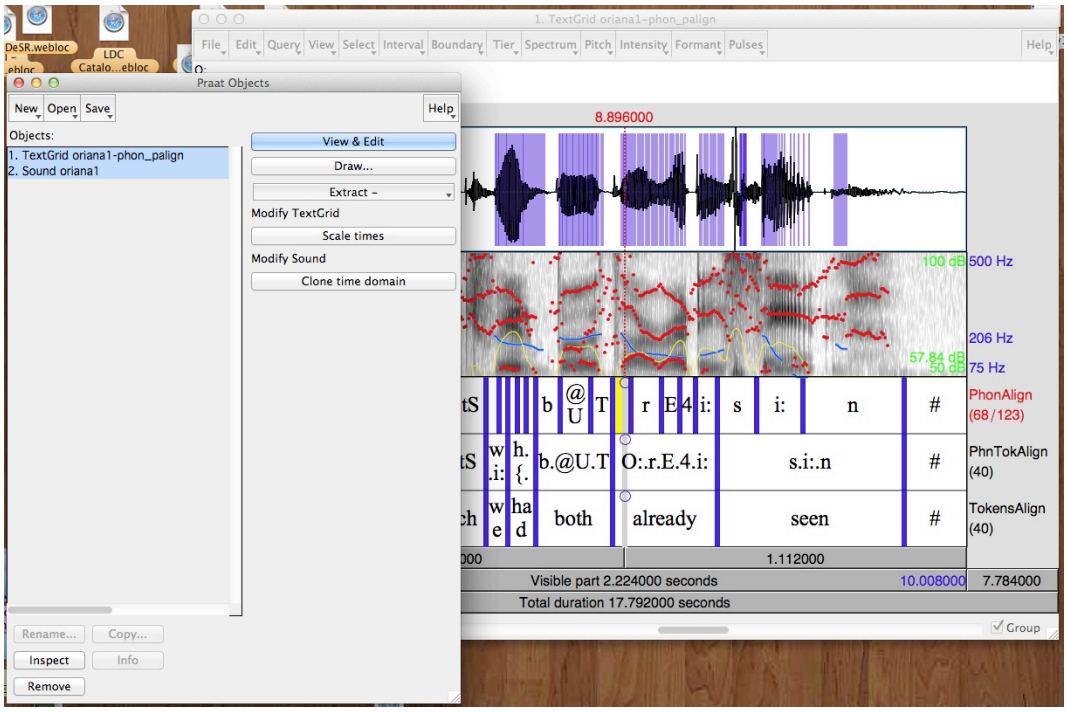

Figure 2 - Praat editing the TextGrig generated by SPPAS

Praat permits to adjust the limits of the intervals, so that errors made by SPPAS can be fixed. Check how SPPAS aligned non-words; tag each non-word audio segment with the token *<now>* (just a symbol to mean *no*n-*w*ord).

Then, use Praat to add more interval tiers to the TextGrid; in particular add the following tiers:

- DialogueAct
- Speaker

The first one will be used to tag entire phrases with the corresponding dialogue act, according to the theory of "dialogue acts" proposed by DAMSL (see the two documents on DAMSL).

The second one tags entire utterances belonging to a given speaker.

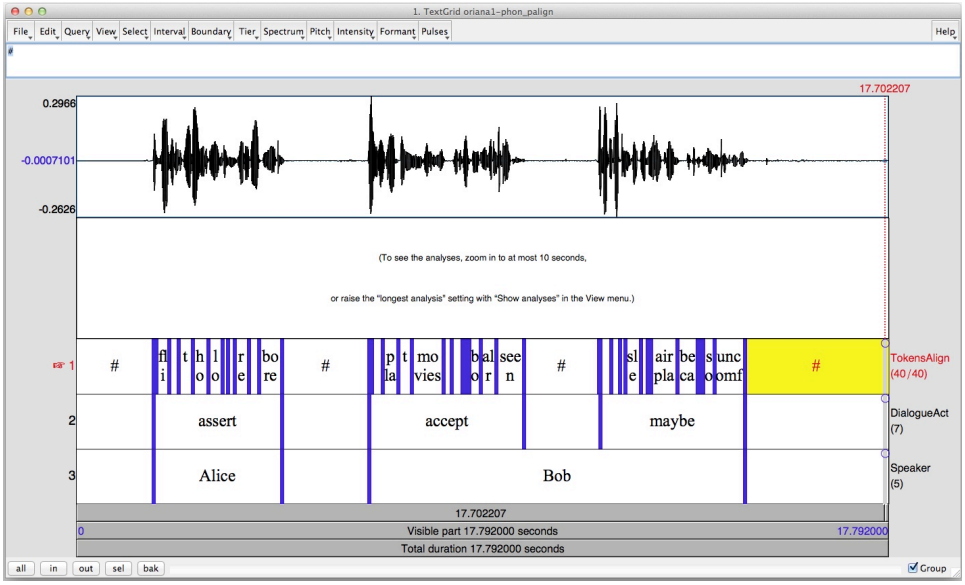

Figure 3 - Adding tiers to the TextGrid

Save the resulting TextGrid as *dialogue.TextGrid*

2.3 Adding audio features

Write a Praat script and calculate, per each token in the TokensAlign tier of the *dialogue.TextGrid* file, the average pitch and the average intensity.

The result of this calculation should be a txt file, named *dialogue-audio.txt* where each token is associated with its average pitch and intensity. You could also add information given by the other tiers. For example[2]:

Token	*Pitch*	*Intens.*	*Speaker*	*DialogueAct*
how	933	59	Alice	info-request
are	1213	51	Alice	info-request
you	1023	48	Alice	info-request
fine	954	47	Bob	answer

2.4 Adding textual features

Pass to a POS tagger the content[3] of the *dialogue.txt*, excluding the speaker names. Then, add to each token in the *dialogue.txt* file the related POS; the result is the *dialogue-audio-pos.txt* file:

[2] It's just an example, you could use whatever format you prefer, even XML-based if you like it.

[3] Probably you should pass to the POS tagger the content of the file, line by line.

Token	*Pitch*	*Intens.*	*Speaker*	*DialogueAct*	*POS*
how	933	59	Alice	info-request	WRB
are	1213	51	Alice	info-request	VBP
you	1023	48	Alice	info-request	PRP
fine	954	47	Bob	answer	JJ

2.5 Adding some "semantics"

To add some, admittedly very simple, semantic information to your data set, try to leverage WordNet and its *domains*: for each noun, verb, adjective, and adverb in the *dialogue-audio-pos.txt* file, write a piece of Java code that, using JWNL, searches a pair (word, POS) in the WordNet database, extract all the synsets, and get all the domains associated to such synsets. If, no domain is associated to any synset, try with the hypernym synsets until you reach the root.

Then save a file, named *dialogue-audio-pos-domains.txt*, where each tokes is associated with the list of its domains (use a special string, as *<none>* for tokens without domains):

Token	*Pitch*	*Intens.*	*Speaker*	*DialogueAct*	*POS*	*Domains*
how	933	59	Alice	info-request	WRB	<none>
are	1213	51	Alice	info-request	VBP	<none>
you	1023	48	Alice	info-request	PRP	<none>
fine	954	47	Bob	answer	JJ	<nome>
...						
light	1234	55	Bob	offer	NN	physics#1, natural philosophy#1

2.6 Showing the results

The file *dialogue-audio-pos-domains.txt* contains all the information you need for generating some interesting statistics and diagrams about the dialogue[4]:

- Prepare two graphs showing how the pitch and intensity varies over the tokens of the whole dialogue. In such graphs, be sure that utterance of each speaker is recognizable (you could, for example, change the graph color for distinguish the segments belonging to the first or to the second speaker).
- For each speaker: for each dialogue act, calculate average pitch and average intensity; then, prepare two graphs showing how the pitch and intensity varies over such dialogue acts.
- For each speaker, show how many dialogue acts she/he uttered, divided by dialogue act typology (for example, Alice uttered 3 Info-request, 2 Answer, etc.)
- For each speaker, calculate the total number of non-words; calculate the number of non-words for each dialogue act typology (for example, sum the number of non-words for all the "info-request" dialogue acts).

[4] you could use Excel or other tools to prepare stats and diagrams.

- For each speaker: for each dialogue act typology, calculate the fraction of tokens belonging to a given dialogue act typology, with respect to the total number of tokens uttered by the speaker.
- For each speaker, show how many WordNet domains are associated to her/his utterances; and show how many tokens belong to such WordNet domains.

3 How to write the Report

You can write the report in Italian or English. The report should contain:
1. A brief description of the problem to solve, and a brief intro on the NLP tools.
2. A brief description of what you did.
3. Results you obtained (accuracies, charts).
4. Discuss the results (e.g. how the indexes depends on the test set size, etc.)

Remember to provide all the binary, txt, and scripts files you generated during the process.

4 Info

Problems with the tools? Typos, mistakes or unclear parts in this text? Contact *roberto.tedesco@polimi.it*

5 Deadline

Deadline: at least a week before your oral exam.
Send an e-mail, attaching the report and all the related files, to: *licia.sbattella@polimi.it* and *roberto.tedesco@.polimi.it*
The report will be discussed as a part of the oral exam.

Good work!

Tools

Suggested tools (in random order…).

POS tagger

TreeTagger (POS tags & word base forms)
http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/

Stanford POS tagger
http://nlp.stanford.edu/software/tagger.shtml

FreeLing
http://nlp.lsi.upc.edu/freeling/

Audio processing

Praat
http://www.fon.hum.uva.nl/praat/

SPPAS
http://aune.lpl.univ-aix.fr/~bigi/sppas/

WordNet

WordNet
http://wordnet.princeton.edu

JWNL
http://sourceforge.net/apps/mediawiki/jwordnet/index.php?title=Main_Page