SLSeg Readme*

Milan Tofiloski, Julian Brooke, Maite Taboada

1 Platforms Supported

Unix-based platforms, Mac OS X.
Please refer to the COPYING.txt file for license information.

2 Software Required and Dependencies

These softwares require the installation of python 2.4 or 2.5 (tho it might work with other versions since we haven't tested every possible permutation), and perl 5.

python is available at: http://python.org/

perl is available at: http://www.perl.org/

You must also install nltk 0.9.5 after you have installed python, available from (not compatible with later versions of nltk yet): http://www.nltk.org/

breakSent-multi.pl is included. It was used in the Document Understanding Conference (2003 and beyond). It was made available by NIST, but there currently is no active link to the software (there was also no license information attached with the script). Place the contents of the duc2003.breakSent folder in the same directory as the included python scripts (see included software below as well as the screenshot illustrating the directory structure).

Charniak's parser is not included with SLSeg, and needs to be downloaded. It is available for dowload from:
http://www.cs.brown.edu/people/ec/#software
The zipped file to dwld is called 'parser05Aug16'.

You can install parser05Aug16 and duc2003.breakSent folders anywhere in your system, but the paths in these instructions assume you place the folders all within the same directory. An example directory layout is illustrated

*This work was supported by an NSERC Discovery Grant (261104-2008) to Maite Taboada.

in the attached screenshot, **my_final_directory_structure_after_installation.png**. You do not need to have yours identical, but will have to modify your paths if the layout is not identical.

To install Charniak's parser, type 'make parseIt' from the 'parser05Aug16/PARSE' directory.

3 Included Software

This package comes with the 5 python files and a few text files that contain data which the python scripts use. The python files are:

- swap.py
- insert_paragraph_breaks_for_breakSent.py
- prep_as_input_to_charniak_parser.py
- segmenter.py
- run_all.py

The data files included are:

- attributionverbs.txt
- badpreplist.txt
- ifverblist.txt
- toverblist.txt
- surfacebreakrules.txt
- phrasaldiscoursecues.txt

4 Description of Included Scripts

The run_all.py script executes 6 programs:

- swap.py
- insert_paragraph_breaks_for_breakSent.py
- breakSent-multi.pl

- prep_as_input_to_charniak_parser.py
- parseIt
- segmenter.py

A short description of the purpose of each script follows.

4.1 swap.py

input: raw text file
output: file with all contractions expanded (used by the parser)

4.2 insert_paragraph_breaks_for_breakSent.py

input: raw text file with expanded contractions
output: text file with inserted paragraph breaks <PARAGRAPH> </PARAGRAPH> since breakSent inserts carriage returns after each sentence

4.3 breakSent-multi.pl

input: raw text file with expanded contractions and paragraph boundaries
output: text with sentence boundaries <s> </s> inserted

4.4 prep_as_input_to_charniak_parser.py

input: text with sentence boundaries <s> </s> inserted
output: text with <PARAGRAPH> tags identified as sentences are removed

4.5 parseIt

input: text with sentence boundaries <s> </s> inserted
output: part-of-speech tags and syntactic tree for each sentence is constructed

4.6 segmenter.py

input: text with POS and syntactic tree of each sentence
output: text segmented into clauses <c> </c>, sentences, and paragraphs

5 Running SLSeg

Assuming the directory structure in the screenshot is followed, and using the included sample directory as input, here is the command to run SLSeg:

```
python run_all.py ./test_samples ./output_test ./parser05Aug16 -T50
```

In general, here is a description of each argument:

```
python run_all.py input_directory output_directory
     path_to_Charniak_parser parseIt_option
```

The parseIt_option to the Charniak parser has a default of -T210. -T50 decreases parse accuracy by 1% but parses 6 sentences/second (whereas -T210 does 1.4 sentences/second).

The output files for each stage are located in the output directory. The final segmented files are located in the step 5 folder.

6 Description of Data Files

The data files are all used by the segmenter.py file. A description of each is provided below. Feel free to modify and add to the files as needed. The two files phrasaldiscoursecues.txt and surfacebreakrules.txt are not merely just lists of individual words; they require special notation to handle the fact that they are multi word expressions and also depend on syntactic context (e.g., whether or not a phrase precedes an auxilliary). These are lexical rules that are also partly designed to compensate for incorrect parses due to statistical parsers.

6.1 attributionverbs.txt

This contains a list of attributive verbs that should not be considered the main verb in a discourse unit since they signal an attributive or cognitive state which does not have a rhetorical relation between it and its attributive statement, as in the example *I think John went to the store*. This sentence is a single unit and not two independent units such as *I think* and *John went to the store*.

6.2 badpreplist.txt

This contains a list of prepositions such as *of, in, with, into, above, to*, etc. which are used as complements to restrict when they can be used

as markers for signaling discourse units. The sentence *He was the king of playing jazz* should be a single unit.

6.3 ifverblist.txt

Contains a list of words that typically are used in conjunct with the words *if* and *whether*, such as *forget, prove, doubt, assume*, etc. and should not be treated as a separate discourse unit. The sentence *Electing the Republicans makes one consider whether the voting was fair* should be treated as a single discourse unit, whereas the sentence *Electing the Republicans to a second term would cause a disaster whether the population voted fairly or not* should have a discourse boundary inserted between *disaster* and *whether*.

6.4 toverblist.txt

This is similar in purpose to the *ifverblist.txt* file. Words contained in this file are typically found to precede the word *to* and should not have discourse boundaries placed between them. Such words are *going, try, seem, pretend*, etc. An example is *The man carrying wood is going to build a fire*, which should not have a discourse boundary placed between *going* and *to*.

6.5 phrasaldiscoursecues.txt

This is a list of phrases that technically satisfy the syntactic requirements for a break, but function as discourse markers rather than separate semantic units. SLSeg will avoid breaking at the edge of these phrases.

Words are separated with an underscore _. Alternatives are separated with a |. The least common word with no alternatives should be marked with @ (the 'head').

For example:

You know, I don't think this is a very good example.

Based purely on the syntax, we would break after the comma, however, 'you know' is not providing semantic content, but rather serves a discourse function (indicating, for instance, that the proposition is something you have just figured out), so we do not break.

6.6 surfacebreakrules.txt

This is a list of rules that allow for breaks based on surface words or tags. Words (or tags) are separated with an underscore _. Alternatives are separated with a |. The least common word with no alternatives should

be marked with @ (the 'head'). A number before the @ indicates the index where the break should be placed (0 = directly before the head). Optionally, you can place ^*words/tags* before/after the phrase to indicate what cannot come before or after.

An example, for the phrase 'instead of' that is followed by a gerund (VBG), you would write:
0@instead_of_VBG
to handle the sentence:
I learned to play the piano
instead of learning to play baseball.

7 Description of Segmentation Tags

The following describes what each tag pair used in the output file signifies.

- <T> </T> - these tags capture the entire text; they delineate where the text begins and ends
- <P> </P> - these tags mark paragraph boundaries
- <s> </s> - these tags mark sentence boundaries
- <c> </c> - these tags mark clause boundaries
- <M> </M> - M is used to mark the trace of a NP which has been moved elsewhere to preserve discourse unit constituency. They should be ignored for the purposes of discourse parsing, but preserved for readability.
- <breakWithinParens> - these tags mark clause boundaries that occur within parentheticals (parenthesis, matching colons and matching hyphens)

8 References

Tofiloski, M., J. Brooke and M. Taboada. *A Syntactic and Lexical-Based Discourse Segmenter*. 47th Annual Meeting of the Association for Computational Linguistics. Singapore, August 2009.