HW #5

Goals

Through this assignment you will:

- Explore the role of features in implementing linguistic constraints.
- Identify some of the challenges in building compact constraints to define a precise grammar.
- Gain some further familiarity with NLTK.
- Apply feature-based grammars to perform grammar checking.

Background

Please review the class slides and readings in the textbook on feature-based grammars and parsing. Also, review Chapter 9 (Links to an external site.) of the NLTK book for additional detail on feature structures and feature-based parsing in NLTK. A discussion of aspect, relevant to the last few test sentences can be found in J&M 17.4.2

NOTE: *The NLTK book contains a discussion of HPSG-style handling of subcategorization. However, this framework is ***NOT*** implemented in NLTK as it stands. An analogous list structure using [FIRST=?a,REST=?b] pseudo-lists can achieve the same effect, but this should be considered an extra-credit option to be explored if you have spare time. It is not required for this assignment.*

Building a Feature-based Grammar based on the materials above, create a set of context-free grammar rules augmented with features in the NLTK .fcfg format that are adequate to analyze a small set of English natural language sentences. Sample grammars may be found in the NLTK Book Chapter 9 text, in the mini example file referenced below, and in some of the NLTK grammars under /corpora/nltk/nltk-data/grammars. The grammar should be loadable with nltk.data.load().

Your grammar should be able to parse all well-formed sentences in the test sentence file and reject all ill-formed sentences in the list.

Parsing

Create a program to parse the example sentences based on your grammar and analyze the results. Specifically, your program should:

- Load your grammar.
- Load the test sentences.
- For each sentence, you program should output to a file:

- Use nltk.parse.FeatureEarleyChartParser (or your own or similar available feature-based parser) to parse the sentence.
- If the sentence is grammatical and parses, print a single output parse on a single line. You may use the nltk.Tree._pformat_flat function to get single-line output.
- If the sentence is not grammatical and fails to parse, print a single blank line as output.

Note: If the sentence is ambiguous, you only need to print a single parse.

Programming

Create a program called hw5_parser.sh which performs the feature parsing grammar check described above invoked as:
hw5_parser.sh <input_grammar_filename> <input_sentence_filename> <output_filename> where,

- <input_grammar_filename> is the name of the file holding the feature-based grammar that you created to implement the necessary grammatical constraints.
- <input_sentence_filename> is the name of the file holding the sentences to test for grammaticality and parse.
- <output_filename> is the name of the file to write the results of your grammaticality parsing test.

Files

Please adhere to the naming conventions below:

Example and Test Data Files

All data and example files may be found in **/dropbox/20-21/571W/hw5/**.

- **sentences.txt**: Test set of basic sentences to analyze.
- **sentences_key.txt**: Same set of sentences, but marked for acceptability. “*” indicates unacceptability.
- **example_grammar.fcfg**: Toy grammar file in NLTK format with features.
- **example_sentences.txt**: Sentence file to be checked with the example grammar.
- **example_sentences_key.txt**: Same set of example sentences, but marked for acceptability. “*” indicates unacceptability.
- **example_output.txt**: Formatted output file consistent with running the acceptability check/parse on the example sentence file above.

Submission Files

- **hw.tar.gz**: Tarball containing:

    - **hw5_parser.sh**: Primary program file with language-appropriate extension.
    - **hw5_feature_grammar.fcfg**: This file should contain the grammar rules with feature augmentations required to parse the acceptable sentences in the test set and reject the ungrammatical ones. The file should be consistent with the NLTK .fcfg format.
    - **hw5_output.txt**: The output file with the results of parsing each of the input sentences in sentences.txt with your hw5_feature_grammar.fcfg.
- **readme.{txt|pdf}**: Write-up file
    - This file should describe and discuss your work on this assignment. Include problems you came across and how (or if) you were able to solve them, any insights, special features, and what you learned. Give examples if possible. If you were not able to complete parts of the project, discuss what you tried and/or what did not work. This will allow you to receive maximum credit for partial work.

- ## Handing in your work

  All homework should be handed in using the Canvas submission tools.