The University of Texas at Dallas
CS 6320
Natural Language Processing
Spring 2019
Instructor: Dr. Sanda Harabagiu
Grader/ Teaching Assistant: Ramon Maldonado

Homework 2: 100 points (50 points extra-credit)
Issued March 25, 2019
Due April 15, 2019 before midnight

PROBLEM 1: Parsing with your Cocke-Kasami-Younger (CKY) parser (**50 points**)
Generate an automatic CKY parser for the following sentences:
S1: Sales of the company to return to normalcy.
S2: The new products and services contributed to increase revenue.
S3: Dow falls as recession indicator flashed red and economical worries continue through the month.
S4: Figure skater lands historic quadruple jump in senior international competition at the 2019 World Figure Skating Championships on Day 3 but could only clinch a silver medal.

You should generate a grammar for all non-terminals as well as all lexical terminals in the four sentences you need to parse. Convert by hand your grammar to Chomsky Normal Form (CNF) (**10 points**).
Write a program that should do the following:

1. Load the CNF grammar.
2. Read in the example sentences,
3. For each example sentence, output to a file:
    - the sentence itself
    - the simple bracketed structure parse(s) based on your implementation of the CKY algorithm, and
    - the number of parses for that sentence.

Running your code should be performed from:
Hw2_CKYparser.{py|java|etc} <grammar_filename> <sentence_filename> <output_filename>
where:

- <grammar_filename> is the name of the file holding grammar rules in the in Chomsky Normal Form.
- <test_sentence_filename> is the name of the file containing four test sentences to parse with your algorithm.
- <output_filename> is the name of the file where your system will write the parses and over the test sentences.

You will receive 10 points for each sentence that is automatically parsed by your program.

PROBLEM 2: Statistical and dependency parsing (**25 points**)
In this problem you asked to use several existing implementations of statistical, neural and dependency parsers. Namely, you should use:

a] Charniak and Johnson's BLLIP Parser, described in:

```
Charniak, Eugene, and Mark Johnson. "Coarse-to-fine n-best parsing
and MaxEnt discriminative reranking." Proceedings of the 43rd Annual
Meeting on Association for Computational Linguistics. Association
for Computational Linguistics, 2005.
Available at: http://www.aclweb.org/anthology/P05-1022
```

- You can use the NLTK interface to this parser!!!

b] Constituency Parsing with a Self-Attentive Encoder (Kitaev & Klein, 2018), described in the paper available at:
http://www.aclweb.org/anthology/P18-1249

CODE available at:
https://github.com/nikitakit/self-attentive-parser

c] Stanford Dependency parser described at:
https://nlp.stanford.edu/software/lex-parser.html

d] Stanford Neural Network Dependency parser described at:
https://nlp.stanford.edu/software/nndep.html

1] You will produce manually the dependency parse of the four sentences used in Problem 1, indicating at each step: (a) the stack; (b) the word list; (c) the action (labeled with the dependency relation) and (d) the relation added to the parse. (**8 points**)

2] You will execute both Stanford dependency parsers on all four sentences and discuss which of them performed best on each of the sentences, and why. You will show the results of the parses on each of the sentences. (**9 points**)

3] You will execute the two constituency parsers (BLIPP and Kitaev & Klein, 2018) to enable you to discuss what errors you observe that the BLIPP parser of the Constituency parser with self-attentive encoding made on any of the four sentences. You will show the results of the parses on each of the sentences. (**8 points**)

PROBLEM 3: Semantic Role Labeling (**25 points**)
1] Considering the same four sentences as in Problem 1, identify by hand the predicates and their arguments against ProbBank definitions, available at:
http://verbs.colorado.edu/verb-index/index.php
Indicate clearly the span of the arguments on each sentence, by marking them as:
Sales of [the company]$_{Arg???}$ *to return to normalcy.*
(**15 points**)

2] Perform automatic semantic role labeling on the same four sentences using the neural SRL reported in "*Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling",* available from:
https://aclweb.org/anthology/P18-2058

CODE available from:
https://github.com/luheng/lsgn
Discuss the differences you observe from your manual parses. (**10 points**)

Software Engineering (includes documentation for your programming assignments)

Your README file must include the following:

- Your name and email address.
- *Homework number* for this class (NLP CS6320), and the *number of the problem* it solves.
- A description of every file for your solution, the programming language used, supporting files, any NLP tools used, etc.
- How your code operates, in detail.
- A description of special features (or limitations) of your code.

Within Code Documentation:

- Methods/functions/procedures should be documented in a meaningful way. This can mean expressive function/variable names as well as explicit documentation.
- Informative method/procedure/function/variable names.
- Efficient implementation
- Don't hardcode variable values, etc

EXTRA-CREDIT PROBLEM 1 (30 points):

The main goal of the extra-credit problem 1 is test your ability to manually produce the CCG parse of the sentences:

S1: Sales of the company to return to normalcy. (**3 points**)

S2: The new products and services contributed to increase revenue. (**5 points**)

S3: : Dow falls as recession indicator flashed red and economical worries continue through the month. (**8 points**)

S4: Figure skater lands historic quadruple jump in senior international competition at the 2019 World Figure Skating Championships on Day 3 but could only clinch a silver medal. **(14 points)**

EXTRA-CREDIT PROBLEM 2 (20 points):

The main goal of the extra-credit problem 2 is to test your ability to perform semantic parsing based on FrameNet. To access the FrameNet index, go to:
https://framenet.icsi.berkeley.edu/fndrupal/frameIndex

1/ Annotate by hand the Frames and corresponding Frame Elements(FEs) in each of the sentences used in Problem 1. Mark clearly the text span of the FEs. (**10 points**).

2/ Use the neural semantic labeler operating based on FrameNet data, available at:
https://github.com/microth/mateplus
to automatically recognize the frames and frame elements on the four sentences used in Problem 1. Discuss the differences from your manual annotation. (**10 points**).