

The University of Texas at Dallas
CS 6320
Natural Language Processing
Spring 2020
Instructor: Dr. Sanda Harabagiu
Grader/ Teaching Assistant: Ramon Maldonado

Homework 2: 100 points (215 points extra-credit)
Issued March 11, 2020
Modified March 23, 2020
Due April 13, 2020 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 parsing (35 points)

In this problem you asked to use several existing implementations of statistical and neural 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>

1] You will execute one of the two constituency parsers (BLIPP or Kitaev & Klein, 2018) of your choice.

You will show the results of the automatic parse on at least one of the sentences from Problem 1. You will receive **10 points** for any automatically parsed sentence. If you discuss the differences (errors) against the parse you obtained in Problem 1 for the same sentence, you receive **5 points. (15 points)**

Extra-credit: If you produce automatic parses any additional sentence, you receive **10 points** per parsed sentence. If you discuss the errors you observe against the parses you have generated in Problem 1 you will receive 2 points per sentence. **(38 points)**

3] You will manually recognize all the Noun and Verb Phrases in the following two sentences:

For more than 30 years, Mauro Morandi has been the sole inhabitant of a beautiful island in the Mediterranean Sea. For the past few weeks his hermit's hut has been an aptly isolated location from which to watch the global coronavirus crisis unfold.

You will indicate the NP premodifiers, postmodifiers as well as the heads of the NPs and VPs. **[10 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 PropBank 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.

You should perform the manual semantic role labeling with PropBank for at least one of the sentences. **(10 points)**

Extra-credit: **(30 points)** If you generate the dependency parse for more than a sentence, you will receive an additional **10 points** per sentence.

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>

You should perform the automatic semantic role labeling with PropBank for at least one of the sentences. **(10 points)**

Extra-credit: **(30 points)** If you generate the automatic role labeling with PropBank for more than a sentence, you will receive an additional **10 points** per sentence.

3] Discuss the differences you observe from your manual parses of the sentence you have processed. **(5 points)**

Extra-credit: **(15 points)** If you discuss the differences from the manual labeling for more than a sentence, you will receive an additional **5 points** per sentence.

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 (72 points):

The main goal of the extra-credit problem 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 four sentences used in Problem 1. Mark clearly the text span of the FEs. You will receive 3 points per each sentence correctly labeled. **(12 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. You will receive *10 points* per sentence that is labeled with FrameNet semantic role automatically. **(40 points)**. Discuss the differences from your manual annotation. You will receive 5 points per sentence discussed **(20 points)**.