Assignment 3

Mingyang Zhou

UCDAVIS
UNIVERSITY OF CALIFORNIA

Assignment 3

- Instructions are on canvas
- Download the scripts and data from canvas
- Due Mar 22 at 11:59pm

Intro: Scripts and Data

- cfggen.pl randomly generate samples from a PCFG
- cfgparse.pl parses sentences using a PCFG by finding the most probable parse
- grammar1 , grammar2 , and lexicon give you the starting point for building a PCFG for a subset of English
- Examples.sen gives you some sentences in that subset of English

Intro: Scripts and Data

./cfggen.pl --text <N> grammar1 lexicon

```
kevin@kevin-ThinkPad-T540p:~/Kevin/UC_Davis/WA2018/ECS189_AI_NLP$ ./cfggen.pl --text 10 grammar1 lexicon
1:  that chalice has any servant .
2:  any husk is Zoot .
3:  the quest is Uther Pendragon .
4:  no weight is a winter .
5:  another land has another fruit .
6:  that quest has no king .
7:  a story has this sovereign .
8:  each servant has any king .
9:  this weight carries no winter .
10:  any husk into every land drinks a king .
```

Intro: Scripts and Data

./cfgparse.pl grammar1 lexicon < examples.sen

```
kevin@kevin-ThinkPad-T540p:~/Kevin/UC_Davis/WA2018/ECS189_AI_NLP$ ./cfgparse.pl grammar1 lexicon < examples.sen
4.761e-06 9.522e-06 0.500 (ROOT (S1 (NP (Proper Arthur)) (VP (VerbT is) (NP (Det the) (Nbar (Noun king)))) .))
7.772e-11 1.554e-10 0.500 (ROOT (S1 (NP (Proper Arthur)) (VP (VerbT rides) (NP (Det the) (Nbar (Nbar (Noun horse
(failure)
(failure)
```

Task 1

- Look at the file grammar2

gramma2

ROOT -> S2 1
S2 -> 1
S2 -> +Det 1
S2 -> +Misc 1
S2 -> +Noun 1
S2 -> +Prep 1
S2 -> +Proper 1
S2 -> +VerbT 1
+Det -> Det 1
+Det -> Det +Det 1
+Det -> Det +Misc 1
+Det -> Det +Noun 1
+Det -> Det +Prep 1
+Det -> Det +Proper 1
+Det -> Det +VerbT 1

Lexicon

Noun -> castle 1
Noun -> king 1
Noun -> defeater 1
Noun -> sovereign 1
Noun -> servant 1
Noun -> corner 1
Noun -> land 1
Noun -> quest 1
Noun -> chalice 1

Task 1

- Run:

./cfgparse.pl grammar2 lexicon < examples.sen

- What kind of language model does this PCFG implement? Give your thoughts.

Task 2

- Look at the file grammar1

gramma1

ROOT -> S1 99
S1 -> NP VP . 1
VP -> VerbT NP 1
NP -> Det Nbar 20

Task 2

- Compare the outcome when you run:

./cfgparse.pl grammar1 lexicon < examples.sen

./cfgparse.pl grammar1 grammar2 lexicon < examples.sen

- Explain what's going on

Task 3

- Compare the outcome when you run:

./cfggen.pl --text <N> grammar1 lexicon

./cfggen.pl --text <N> grammar2 lexicon

./cfggen.pl --text <N> grammar1 grammar2 lexicon

- Explain what's going on

Task 4

Design your own Grammar:

gramma1

ROOT -> S1 99
S1 -> NP VP . 1
VP -> VerbT NP 1
NP -> Det Nbar 20

my gramma

ROOT -> S1 99
S1 -> NP VP . 8
S1 -> VP 1
S1 -> X1 VP 1
X1 -> AUX NP 1
VP -> VP PP 20
VP -> VerbT NP 80
NP -> Det Nbar 20

Task 4

Design your own Lexicon:

Lexicon

VerbT -> carries 1
VerbT -> rides 1
Misc -> ! 1
Misc -> . 1
Misc -> ? 1
Misc -> , 1
Misc -> and 1
Misc -> but 1
Misc -> or 1
Misc -> either 1

My Lexicon

VerbT -> carries 1
VerbT -> rides 1
Punc -> ! 1
Punc -> . 10
Punc -> ? 3
Punc -> , 10
CC -> and 10
CC -> but 2
CC -> or 1
CC -> either 1

Task 4

Constraints:

1. You can only generate binary or unary rule in grammar and lexicon file.

A -> B C
A -> B
A -> B C D

2. Cannot include new words in lexicon as terminals.

Task 4

Goals:

1. The PCFG will predict high probability for a grammatical English sentence and predict low probability for an ungrammatical sentence.

```
./cfgparse.pl mygrammar mylexicon < examples.sen
```

2. The PCFG will never fail to parse a string of words.
3. The PCFG will be able to generate grammatical sentences.

Task 4

Competitive Task

1. Minimum Requirement: (1) Beat the performance of a merged gramma from **gramma1** and **gramma2** with the default **lexicon**. (2) Never fail to parse any sentence in the test dataset.
Meet this requirement to get full points for this task.
2. Compete against you classmates: Top 10% get 5 extra points for this assignment.

Task 4

Evaluation Metrics

- Collect grammatical sentences generated from each of your grammar to form a test dataset.
- Compute the cross-entropy to evaluate how well can your gramma predict the sentences in the test dataset.
- P(s) – The probability of the string s is the sum of the probabilities of the trees which have that string as their yield
- The lower value is better.

$$2^{\frac{-log_2 p(s_1) - log_2 p(s_2) - log_2 p(s_3) ... - log_2 p(s_n)}{n}}$$

Task 5

Evaluation Metrics

- Generate 20 sentences with your grammar and lexicon and keep it in a file
- Remove all the ungrammatical sentences from the 20 sentences and keep it in another file.
- What is the fraction of grammatical sentences generated