CSE 517: Homework
Trees

Due Feb 10 Fri 2017 11:59pm

Submission instructions Submit 1 file on Canvas:

1 **Report:** (`HW.pdf`): Please type most of your answers whenever possible. You can include handwritten answers for some questions, in which case, (1) please write them very neatly so that they are legible, and then (2) scan the handwritten answers and include them into your doc as pictures, so that you can make an electronic submission that includes all answers in one file. If you are having trouble formatting your document, please inform the TAs at least 48 hours before the submission deadline so that we can arrange a solution in time.

1 PCFG Language Models

For a variety of NLP applications such as speech recognition, machine translation, and dialogue, a frequently arising situation is to compare multiple hypotheses of sentence realization and choose the most likely option based on their probabilities. We want to know, for example, P(*"I ate a Cherry"*) $\geq$ P(*"Eye eight a Jerry"*). While n-gram language models are the most popular choice:

$$P_{seq}(x) = \prod_i P(x_i | x_{i-2}, x_{i-1})$$

another viable option is to use PCFG as language models. That is, given a sentence x, we can sum over all possible tree structure t whose yield is exactly x:

$$P_{tree}(x) = \sum_{t \in \mathcal{T}(x)} P(t) = \sum_{t \in \mathcal{T}(x)} \prod_{\alpha \to \beta \in t} p(\alpha \to \beta)$$

But computing above can be a tad demanding as you'd need to implement your favorite inside – outside algorithm. So, one might as well approximate this by the maximum $p(t)$, so that,

$$P_{tree}(x) \approx \max_{t \in \mathcal{T}(x)} P(t)$$

While it's a crude approximation, it will be effective enough to tell apart the difference between *"Colorless green ideas sleep furiously"* and *"Green furiously ideas colorless sleep"*. This sort of tree-based language models have shown to be useful in a variety of context, e.g., for machine translation [3], for authorship attribution of text [4], and even for authorship attribution of code [1] [2].

1.1 Deliverables

- (1pt) One evening, Yiming Chua and Xiao Lin go into an intense argument. Xiao believes that $P_{tree}(x)$ is superior to $P_{seq}(x)$ since $P_{tree}(x)$ encodes much more linguistic information than $P_{seq}(x)$. Yiming adamantly disagrees. Finally, they come to you to resolve the issue. What will you say? Your write-up must provide careful and compelling reasons in support of your answer.

– Some issues you can think about concern practical or engineering limitations, while other issues will concern theoretical limitations. Full scores will be given if you address them both. If possible and applicable, provide examples that would illustrate your point clearly.

2 Playing with Off-the-shelf Parsers

In this part of the assignment, you will play with state-of-the-art PCFG and dependency parsers and gain hand-on insights into how modern statistical parsers behave. You can use whatever parsers you want, but here are some options that do include a web interface for your convenience (so that you won't need to download and install anything).

Parsers with web interface:

- `http://tomato.banatao.berkeley.edu:8080/parser/parser.html`
- `http://nlp.stanford.edu:8080/parser/`
- `http://demo.ark.cs.cmu.edu/parse`
- `https://spacy.io/demos/displacy`
- `http://lil.cs.washington.edu/easysrl/demo.cgi`

You might find many of the tags (nonterminals and dependency types) are almost self-explanatory especially if viewed together with examples. As needed, you can also look up the definition of tags from here:

PCFG Nonterminal Tags:

- `http://cs.jhu.edu/~jason/465/hw-parse/treebank-notation.pdf`
- `http://www.surdeanu.info/mihai/teaching/ista555-fall13/readings/PennTreebankConstituents.html`

Dependency Types:

- `http://nlp.stanford.edu/software/dependencies_manual.pdf`

Keep in mind that you can complete this assignment without having to know all these tags in detail.

So, go ahead and try simple sentences like *"my dog ate my homework"* or *"fruit flies like bananas"*. Do the results seem reasonable to you? Try some longer sentences as well, e.g., *"two roads diverged in a wood, I took the one less traveled by, and that has made all the difference"*. You might notice that different parsers operate with slightly different grammar rules or tagging schemes. These details shouldn't matter for the purpose of this assignment.

2.1 Deliverables

- (1pt) Find two different sentences with PCFG parses that you determine as incorrect. Specify which parser you used for those sentences and discuss why you think they are incorrect.

  – Make sure to pick two sentences that show different types of errors. You don't have to identify *all* errors in those sentences. It suffices if you identify *some* errors.

– You must come up with your own unique example sentences. If we find your sentences are identical to your friends', we will invite you to our office for additional exercise.

- (1pt) Similarly as above, find two different sentences with dependency parses that you determine as incorrect. Specify the parsers used and discuss why you think their parses are incorrect.

3 Variations to Cocke–Younger–Kasami (CKY) Algorithm

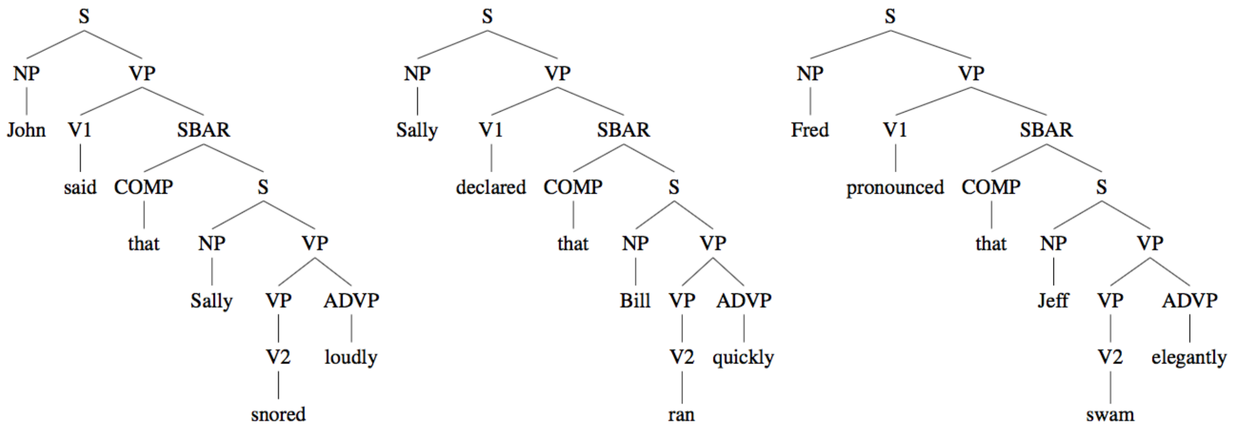
Ashmita Srirasan is not convinced if CFG grammars in the Chomsky Normal Form are ideal for representing the sentence structure as they tend to result in very deep trees. Ashmita decides to experiment with Ashmita Normal Form: in addition to the rules in the form of $X \rightarrow YZ$ and $X \rightarrow w$, Ashmita Normal Form will also allow rules in the form of $X \rightarrow VYZ$, that is, rules that can lead to three nonterminals.

3.1 Deliverables

- (2pts) Write a modified CKY parsing algorithm (as a pseudo code) that can work with CFG in the Ashmita Normal Form.

4 CFG Grammar Refinement

Nathan L. Pedant decides to open a new business and build treebank corpora professionally. After a long agony of creation, he finally produces a corpus which contains the following three parse trees:



Clarissa Lexica then purchases this treebank for \$2.99, and decides to build a PCFG and a parser.

4.1 Deliverables

a (1pt) Show the PCFG (CFG with rule probabilities) that Clarissa would derive from this treebank.

b (1pt) Complete the CKY chart (box cells in the form of half of 6×6 matrix) for parsing the following sentence:

"Jeff pronounced that Bill ran elegantly"

and find all parse trees and their corresponding probabilities.

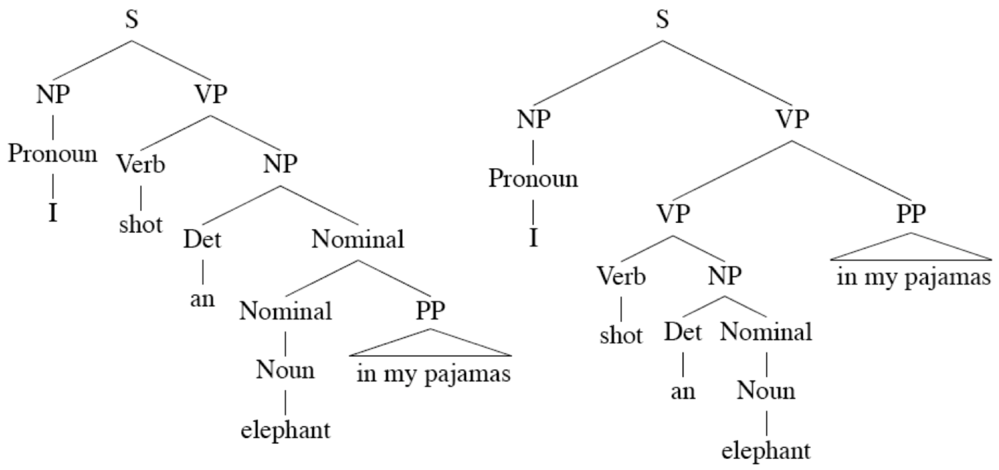
c (2pt) Clarissa is shocked and dismayed that "Jeff pronounced that Bill ran elegantly" has two possible parses, and that one of them—that Jeff is doing the pronouncing elegantly—has relatively high probability, in spite of the fact that such high attachment of ADVP has never seen in the corpus. Since Nathan won't accept returning the treebank, Clarissa decides to fix the treebank herself by *grammar refinement*, i.e., altering some non-terminal labels in the corpus. Show one such transformation to the grammar that would give zero probability to parse trees with "high" attachments.

– **Hint:** consider one of the following options you've seen in the class: *vertical markovization, horizontal markovization, tag split, lexicalization,* or feel free to propose a new idea of grammar refinement.

– While it is not necessary to enumerate all production rules after the grammar refinement, you must (1) spell out at least a subset of the rules with their corresponding probabilities that can clearly demonstrate what systematic changes you are making to the grammar, and (2) explain clearly how that change can ensure zero probability to the high attachment of ADVP.

5 CFG Grammar Refinement II

Clarissa Lexica wants to continue building new treebanks. One experimental treebank Clarissa is contemplating on is a specialized one that focuses on the PP attachment problem:



To address ambiguities such as above, Clarissa is thinking of annotating many more sentences with similar structure and ambiguities, for example,

- I fixed a bug in my code.
- I fixed a bug in my dream.
- I cleaned dishes in my pajamas.
- I cleaned dishes in the sink.
- I ate noodle with tofu.
- I ate noodle with chopsticks.

and so forth.

5.1 Deliverables

- (1pt) Would CKY parsing with the default PCFG, if provided with sufficient amount of annotated trees, learn to disambiguate the correct structure successfully? Explain why or why not. If you chose to answer 'no', then discuss which of the following grammar refinement techniques might have a shot at resolving the ambiguity correctly: *vertical markovization, horizontal markovization, tag split, lexicalization.* Articulate your reasons clearly for full score.

References

[1] Steven Burrows, Alexandra L Uitdenbogerd, and Andrew Turpin. Comparing techniques for authorship attribution of source code. *Software: Practice and Experience*, 44(1):1–32, 2014.

[2] Aylin Caliskan-Islam, Richard Harang, Andrew Liu, Arvind Narayanan, Clare Voss, Fabian Yamaguchi, and Rachel Greenstadt. De-anonymizing programmers via code stylometry. In *24th USENIX Security Symposium (USENIX Security 15)*, pages 255–270, 2015.

[3] Eugene Charniak, Kevin Knight, and Kenji Yamada. Syntax-based language models for statistical machine translation. In *Proceedings of MT Summit IX*, pages 40–46. Citeseer, 2003.

[4] Sindhu Raghavan, Adriana Kovashka, and Raymond Mooney. Authorship attribution using probabilistic context-free grammars. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 38–42. Association for Computational Linguistics, 2010.