Statistical NLP 2019 - Assignment 1

due Monday Sept 9 - at 11:59pm EST by email to aparikh@cs.nyu.edu and urvish@nyu.edu

Deliverables:

- Answers to problem set questions. If you use good handwriting, you are welcome to handwrite them and email us a scanned copy.
- Signed course rules / cheating policy handed in class (On September 10)

Submission Format: Please title your submission email **Assignment 1 submission**, with the report in pdf format (firstname_lastname_hw1.pdf). Write your netID and your collaborators' netID (if any) in the report.

1 Course Rules / Cheating Policy

I understand that most students would never consider cheating. There is, however, a fraction of students for whom this is not the case. To make sure we have a common understanding of what the course rules are, I ask you to print the last page of this assignment and acknowledge the rules by signing it. Please hand in your signed copy in class.

2 Problem Set (45 points)

Consider n-gram language modeling:

$$P(w_i|w_1,...,w_{i-1}) \approx P(w_i|w_{i-n+1},...,w_{i-1}) = P(w_i|w_{i-n+1}^{i-1})$$

Let V denote the vocabulary size (typically around a million), n the n-gram order (typically around 5), and C the number of tokens in the corpus used to compute the relevant counts (typically around 1 billion).

1. **(15 points)** Characterize the memory complexity of Kneser Ney language models in big-O notation. You should decide what variables are important to model (Hint: think about how to store the relevant counts efficiently).

2. **(15 points)** The term *inference time* refers to one call to the language model i.e. computing $P(w_i|w_{i-n+1}^{i-1})$ under the language model for one choice of w_{i-n+1}^{i-1}, w_i. Characterize the inference time complexity of Kneser Ney language models in big-O notation. You should decide what variables are important to model (Hint: Think about which quantities can be cached efficiently to speed up inference time).

3. **(15 points)** Recall absolute discounting for $n = 2$:

$$P_{ad}(w_i|w_{i-1}) = \frac{\max(c(w_{i-1}, w_i) - d, 0)}{\sum_w c(w_{i-1}, w)} + \alpha(w_{i-1})\widehat{P}(w_i)$$

where $\widehat{P}(w_i) = \frac{c(w_i)}{\sum_w c(w)}$ is the empirical unigram distribution, d is the discount and α is the left-over weight assigned to the lower order distribution:

$$\alpha(w_{i-1}) = \frac{d \times |\{w : c(w_{i-1}, w_i) > 0\}}{\sum_w c(w_{i-1}, w)}$$

For $n = 2$ (a bigram model), give an example of a small vocabulary V and corpus C where absolute discounting does not preserve the marginal constraint i.e.

$$\widehat{P}(w_i) \neq \sum_{w_{i-1}} P_{ad}(w_i | w_{i-1}) \widehat{P}(w_{i-1}) \tag{1}$$

3 Course Rules / Cheating Policy

I understand that most students would never consider cheating in any form. There is, however, a fraction of students for whom this is not the case. To make sure we have a common understanding of what the course rules are, I ask you to print this page and acknowledge the rules by signing it. Please hand in your signed copy in class.

The rules below are adapted from Smith & Dyer at CMU (see their class Natural Language Processing (11-{4,6}11)).

- You may verbally collaborate on homework assignments. On each problem and program that you hand in, you must include the names of the people with whom you have had discussions concerning your solution. Indicate whether you gave help, received help, or worked something out together. The names should include anyone you talked with, whether or not they're taking the class, and whether or not they attend or work at NYU.
- You may get help from anyone concerning programming issues which are clearly more general than the specific assignment (e.g., "what does a particular error message mean?").
- You may not share written work or programs (on paper, electronic, or any other form) with anyone else.
- If you find an assignment's answer, partial answer, or helpful material in published literature or on the Web, you must cite it appropriately. Don't claim to have come up with an idea that wasn't originally yours; instead, explain it in your own words and make it clear where it came from.
- On the course project, you are encouraged to use existing NLP tools. You must acknowledge these appropriately in all documentation, including your final report. If you aren?t sure whether a tool or data resource is appropriate for use on the project, because it appears to solve a major portion of the assignment or because the license for its use is not clear to you, or if you aren't sure how to acknowledge a tool appropriately, you must speak with the course staff.

Clear examples of cheating include (but are not limited to):

- Showing a draft of a written solution to another student.
- Showing your code to another student.
- Getting help from someone or some resource that you do not acknowledge on your solution.
- Copying someone else's solution to an assignment.
- Receiving class related information from a student who has already taken the exam.
- Attempting to hack any part of the course infrastructure.
- Lying to the course staff.

I hereby acknowledge that I have read and understood the course rules.

Date:

Name:

Signature: