Department of Computer Science
The University of Melbourne
COMP90042 WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2018)

Workshop exercises: Week 7

**Discussion**

1. For the following "corpus" of two documents:

   ```
   1. how much wood would a wood chuck chuck if a wood chuck
   would chuck wood
   2. a wood chuck would chuck the wood he could chuck if
   a wood chuck would chuck wood
   ```

   (a) Which of the following sentences: `a wood could chuck`; `wood would a chuck`; is more probable, accoding to:
   i. An unsmoothed uni-gram language model?
   ii. A uni-gram language model, with Laplacian ("add-one") smoothing?
   iii. An unsmoothed bi-gram language model?
   iv. A bi-gram language model, with Laplacian smoothing?
   v. An unsmoothed tri-gram language model?
   vi. A tri-gram language model, with Laplacian smoothing?

2. What does **back–off** mean, in the context of smoothing a language model? What does **interpolation** refer to?

3. Name the key differences and similarities between n-gram language models versus the log-bilinear language model and feed-forward neural language model.

4. What does **recurrent** mean in the context of a recurrent neural network (RNN) language model? How does the approach differ from a feed-forward language model?

**Programming**

1. Using the iPython notebook `WSTA_N11_n-gram_language_models`, randomly generate some sentences based on the bi-gram models of the Gutenberg corpus and the Penn Treebank. What do you notice about these sentences? Are there any sentences which might get returned for both corpora? Why?

2. Find a sentence with a higher probability than *revenue increased last quarter.*, according to:

   (a) The Gutenberg corpus, using bi-grams smoothed with Laplacian smoothing
   (b) The Gutenberg corpus, using bi-grams smoothed with Interpolation
   (c) The Penn Treebank corpus, using bi-grams and Laplacian smoothing
   (d) The Penn Treebank corpus, using bi-grams and Interpolation

3. Find the perplexity of the above (smoothed) language models for a number of sentences. Why does Interpolation generally have better perplexity?

4. Follow the `tensorflow` tutorial on RNN language models at `https://www.tensorflow.org/tutorials/recurrent`. Alternatively, PyTorch is another similarly excellent python deep learning library, which also has an RNN tutorial `http://pytorch.org/tutorials/beginner/nlp/sequence_models_tutorial.html`.

**Catch-up**

- What is a **language model**? What is an $n$**-gram language model**? Why are language models important?
- What do **uni-gram**, **bi-gram**, **tri-gram**, etc. signify?
- Why is **smoothing** important?
- Why do we usually use **log probabilities** when finding the probability of a sentence according to an $n$-gram language model?
- How might one evaluate a language model?

**Get ahead**

- Using the (short) "corpus" from Discussion Q1, generate all of the sentences of length 3. Choose an $n$-gram language model, and find the most probable sentence. What about length 4? 5? 6? What do you notice about these sentences? Does smoothing (or not) change this?
- Modify the iPython notebook so that it uses back–off smoothing. How does this change the probability of the given sentence? Why? Is the perplexity of this model better than Laplacian smoothing? Interpolation? Why?
- Perform the Programming experiments above using different corpora.