School of Computing and Information Systems
The University of Melbourne
COMP90042 NATURAL LANGUAGE PROCESSING (Semester 1, 2022)

Workshop exercises: Week 3

Discussion

1. What is **text classification**? Give some examples.

Sentiment Analysis of tweets / Spam Detection / Native Language Identification

① Document Rep: text → Number
② Feature Selection
③ Sparse Data Prob

   (a) Why is text classification generally a difficult problem? What are some hurdles that need to be overcome?

   (b) Consider some (supervised) text classification problem, and discuss whether the following (supervised) machine learning models would be suitable:

   i. k-Nearest Neighbour using Euclidean distance
   ii. k-Nearest Neighbour using Cosine similarity
   iii. Decision Trees using Information Gain
   iv. Naive Bayes
   v. Logistic Regression
   vi. Support Vector Machines

{ X for high-D problems.
tend to prefer rare features
Assumption (feature independent)
Binary classification
→ can't handle multi-classes problems. (label > 2)

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

bigram

```
1. <S> how much wood would a wood chuck chuck if a wood chuck
would chuck wood
2. <S> 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` (A); `wood would a chuck` (B); is more probable, according 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?

Count :

how	much	wood	would	a	chuck	if	the	he	could	<\s>	Total
1+1	1+1	8+1	4+1	4+1	9+1	2+1	1+1	1+1	1+1	2+1	34+11

i. $P(A) = P(a) \cdot P(wood) \cdot P(could) \cdot P(chuck) \cdot P(<\!s\!>)$

$= \frac{4}{34} \times \frac{8}{34} \times \frac{1}{34} \times \frac{9}{34} \times \frac{2}{34} = 1.27 \times 10^{-5}$

ii. $P(A) = \frac{5}{45} \times \frac{9}{45} \times \frac{2}{45} \times \frac{10}{45} \times \frac{3}{45} = 1.46 \times 10^{-5}$

iii. $P(A) = P(a|<\!s\!>) \cdot P(wood|a) \cdot P(could|wood) \cdot P(chuck|could) \cdot P(<\!s\!>|chuck)$

$= \frac{C(<\!s\!>a)}{C(<\!s\!>)} \cdot \frac{C(a\ wood)}{C(a)} \cdot \frac{C(wood\ could)}{C(wood)} \cdot \frac{C(could\ chuck)}{C(could)} \cdot \frac{C(chuck\ <\!s\!>)}{C(chuck)}$

$= \frac{1}{2} \times \frac{4}{4} \times \frac{0}{8} \times \frac{1}{1} \times \frac{0}{9} = 0$

(b) Based on the "corpus", the vocabulary = { a, chuck, could, he, how, if, much, the, wood, would, </s> }, and the **continuation counts** of the following words are given as follows:

- a = 2
- could = 1
- he = 1
- how = 0
- if = 1
- much = 1
- the = 1
- would = 2
- </s> = 1

i. What is the continuation probability of chuck and wood?

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

Programming

1. In the 03-classification notebook, observe how different tokenisation regimes alter the text classification performance of the various classifiers on the given Reuters dataset problem.
   (a) Alter the tokenisation strategy so that it incorporates other stages, for example, punctuation, or stemming/lemmatisation.
   (b) Does performance increase or decrease? Are some classifiers affected more than others? Why do you think that is?
2. Using the iPython notebook 04-ngram, 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?
3. 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
4. Find the perplexity of the above (smoothed) language models for a number of sentences. Why does Interpolation generally have better perplexity?

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

- Adjust the `03-classification` iPython notebook, so that the supervised machine learning model attempts to solve the **multi-class** problem, rather than the **single-class** problem (for `acq`). Does your assessment of the relative utility of the given classifiers change?
- Using the (short) "corpus" from Discussion Q2, 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.