School of Computing and Information Systems
The University of Melbourne
COMP90042 NATURAL LANGUAGE PROCESSING (Semester 1, 2021)

Workshop exercises: Week 12

Discussion

1. What is **Summarisation**?

   - Summarisation the task of distilling information from an input text to capture the key points or messages.
   - There are several variants when we consider the nature of input text and the output summary:
     - Single-document summarisation: input is a single document.
     - Multi-document summarisation: input consists of multiple documents.
     - Extractive summarisation: output summary is created by extracting text from the input.
     - Abstractive summarisation: output summary paraphrases the original content and contain novel sentences.

   (a) What is **log likelihood ratio**, and how is it useful for summarisation?

   - The log likelihood ratio of a word measures its statistical discrepancy between two sources. We can use log likelihood ratio to measure how *salient* a word is to a document by comparing its frequency statistics to that of a background corpus. This is useful for summarisation because it provides an unsupervised method to extract salient words/passages from a document.

   (b) What is the log likelihood ratio of word w in document d if: $F_d(w) = 1$, $F_b(w) = 20$, $N_d = 30$, and $N_b = 4000$, where b denotes the background corpus, F the frequency and N the total number of word tokens.

$$\begin{aligned} p &= \frac{1+20}{30+4000} \\ p_d &= \frac{1}{30} \\ p_b &= \frac{20}{4000} = \frac{1}{200} \end{aligned}$$

As $p_d = \frac{1}{30}$ which is greater than $p_b = \frac{1}{200}$, this indicates that w is a word that is rather prominent (i.e. it is statistically more probable in the document than it typically is), but is it *salient* under its log likelihood ratio?

$$\begin{aligned} \text{LLR}(w) &= -2 \times \log \left(\frac{\binom{30}{1} p^1 (1-p)^{29} \times \binom{4000}{20} p^{20} (1-p)^{3980}}{\binom{30}{1} p_d^1 (1-p_d)^{29} \times \binom{4000}{20} p_b^{20} (1-p_b)^{3980}} \right) \\ &= 2.08 \end{aligned}$$

Since its log likelihood ratio is less than 10 (threshold in the lecture), the answer is no: it isn't quite salient enough.

2. You have an idea of building a **comment generation system for news articles**. Training data can be created by mining news articles and their comments on the web. What are the ethical implications of such application? Discuss.

   - What is the actual application (primary use) of such a system? It's difficult to see why we want to automatically generate comments for news articles.
   - It shouldn't be difficult to see that, if such an application is successfully created, it can be used to create large amount of fake comments and interactions online. A malicious organisation/entity can use this to prevent meaningful discussion online by flooding a platform with automatically generated comments.
   - The data collection process also need to be given more thoughts. We need to check the terms of service of a news publication to see whether we can use their news content and more specifically the user comments for research. Just because they are public content does not immediately imply anyone can mine them.

Programming

1. In the iPython notebook `13-embedding-bias`, we build a sentiment classifier using pre-trained word embeddings, and explore the hidden biases contained in the embeddings.

   - Modify the code to remove stopwords and only consider non-stopwords when computing the mean sentiment of a sentence. Does that change the results?
   - Test Word2Vec pre-trained embeddings and see if they contain similar biases. Pre-trained Word2Vec embeddings can be downloaded here (note: it's 1.5GB in size).
   - Revisit the notebook `10-bert`, where we build a sentiment classifier using pre-trained BERT. Test whether this BERT-based sentiment classifier contains similar biases and explain your observations.