School of Computing and Information Systems
The University of Melbourne
COMP90042
WEB SEARCH AND TEXT ANALYSIS (Semester 1, 2019)

Sample solutions for discussion exercises: Week 7

Discussion

1. What is a **POS tag**?

   - A POS tag is a label assigned to a token in a sentence which indicates some grammatical (primarily syntactic) properties of its function in the sentence.

   (a) What are some common approaches to POS tagging? What aspects of the data might allow us to predict POS tags systematically?

   - **Unigram**: Assign a POS tag to a token according to the most common observation in a tagged corpus; many words are unambiguous, or almost unambiguous.
   - **N-gram**: Assign a POS tag to a token according to the most common tag in the same sequence (based on the sentence in which the token occurs) of n tokens (or tags) in the tagged corpus; context helps disambiguate.
   - **Rule-based**: Write rules (relying on expertise of the writer) that disambiguate unigram tags.
   - **Sequential**: Learn a Hidden Markov Model (or other model) based on the observed tag sequences in a tagged corpus.
   - **Classifier**: Treat as a supervised machine learning problem, with tags from a tagged corpus as training data.

   (b) POS tag (by hand) the following sentence: `Pierre Vinken, 61 years old, will join the board as a nonexecutive director Nov. 29.` according to the Penn Treebank tags. (Note that some of the tags are somewhat obscure.)

   - According to the Penn Treebank:

```
NNP Pierre
NNP Vinken
  , ,
 CD 61
NNS years
 JJ old
  , ,
 MD will
 VB join
 DT the
 NN board
 IN as
 DT a
 JJ nonexecutive
 NN director
```

```
NNP Nov.
 CD 29
  . .
```

2. Name the key differences and similarities between n-gram language models versus feed-forward neural language models.

   - n-gram language models and FFNNs share the same setup of using a Markov chain.
   - They factorise the probability of a sentence into the probability of each word given the $n-1$ previous words. The models differ in how this word-based probability model (classifier) is formulated.
   - n-gram models can be considered a "feature-based" model where every ngram is a feature, with a corresponding weight (thus the "weights" for a bigram models form a matrix, for a trigram model a 3d tensor etc.)
   - FFNNLM use an embedding and un-embedding step, to limit the model size and force generalisation (e.g., to locate synonymous words near in vector space), along with more complex functions to couple context to the next word.
   - Another key difference between n-grams and FFLM is that n-grams work over highly sparse data (1-hot word vectors, sparse parameter matrices where more entries are 0), while FFLM work over dense representations

3. 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?

   - Recurrent means that model is structured such that it can be repeatedly applied for each item in a sequence. The recurrence in a RNN is over the hidden states, such that each as new input is fed into the model, this is used to formulate a new hidden state – as a non-linear transformation of the last hidden state, and the new input. In this way the approach can (in theory) represent long-distance phenomena in the sentence, over variable distances.
   - A FFLM instead assumes a fixed sized context, which can be applied using a "sliding window" over a sequence. The hidden state is a function of the n-1 inputs. There is no reuse of computation from previous applications to the sequence.

4. What advantage does a RNN language model have over a feed-forward language model?

   RNNLM can capture long-distance dependencies, while FFLM cannot. For example, it can balance quotes and brackets over long distances.