Com Sci 263

Lecture 3

Quick Recap of W2V skipgram

Outline(Today’s content):

Introduction, intuition and applications of language modelling

From ngrams LMs to feedforwards networks

Language models are regressive, they take previous words and use that input to predict the next words. (Given left hand side context, what is the right hand side context)

P(w) = P(w1 w2 w3…. wk ) =  ∏  P(wk | w1w2w3….wk-1)

What is an N-grams: a contiguous sequence of n tokens from a given piece of text.

A bigram breaks down texts into two tokens each. Thus This is Krish is broken into This is and is Krish. Similar logic for a trigram. N grams LM models p(x{t:t+n}) or p(xt+n | x{t:t+n-1} ). You can get this value from , say, the frequency model, by calculating the frequency of the word occurring in sequence.

When you break down a sequence from documents etc, you would add a eos token (end of sequence) to signify that the sequence ended. You also have a beginning of sequence token(bos).

P(w) = ∏ P(wi|wi-2 wi-1)

Probability of incomplete sentences would be low as per our model, and this this eos would indicate that it is not a good sentence(done by the specific model we are using for the LM)

P(feeds |<bos>,<bos>) should be low as well

Another alternative-> We model the entire sentence and document as a single sequence, and have separate tokens for punctuation. (Standard practice for LLM training)

Using capitalization in LLMs:

By using lowercase, you remove many redundant and overlapping information , however you miss on certain topics such as proper nouns , highlights etc. Reduces the vocab size.

Word Probabilities are really small, and this falls when you multiply these together. Thus, to handle this, instead of multiplying the probabilities we sum the log probabilities.

Breaking down sentences into tokens and comparing the similarity between the embeddings -> problem with this is that is highly dependent on the hyperparams like the n value for n-grams.