Natural Language Processing Tasks with Language Models

I discuss about BERT and N-gram language models which I studied over two months and list my learnings and evaluations about them in a concise manner.

Language Models answer the question: How likely is a string of English words good English? They are used in Speech recognition, Name/Identity recognition, Spelling correction, etc. Their main function is to predict the word/sentence that comes next in a given sequence from a collection of a dataset. For example, we can use conditional probability to design a language model. What is the Probability(Sahil | I am) ?
Now, we as humans have read several texts and gained intuition of how a particular language uses grammar. Similarly, we need to train our model by providing a data set and then testing it on an unseen data set which can help us evaluate the results.

N-gram Language Model

N-gram means "The sequence of N words".
For example, bigrams mean the sequence of two words. What is the Probability(house | That) ? These models are used in suggestions in messengers, spelling correction, machine translation, speech recognition, handwriting recognition, etc.
N-gram takes in many tokens and predicts one token at a time. It works on the property of probability distribution.
For example, there are 4 elements to be predicted and N-gram receives the first element. Now N-gram reads the first element and predicts the second which is equivalent to Probability(am | I). Second step is to find the Probability(a | I am) and finally Probability(programmer | I am a).

BERT Language Model

BERT stands for "Bidirectional Encoder Representations from Transformers". It is a technique for NLP pre-training developed by Google. This model is faster than the previous Long Shot Term Memory (LSTM) models as it has the capacity to take in words simultaneously. BERT is deeply bidirectional, and hence the true context is preserved and understood by it.
BERT can be used to solve tasks such as Neural Machine Translation, Question Answering, Sentiment Analysis and Text Summarization. A key thing to note here is to solve all of the tasks, understanding the language is very important for BERT!

- Training Phase 1: Pre-training BERT to understand language.
  In this phase BERT uses the techniques such as Mask Modelling and Next Sentence Prediction. At the end of this phase, BERT gets a good understanding of the language.

- Mask Modelling Language:
  During this process BERT takes a paragraph with masked tokens that need to be predicted. This is the time during which it learns to predict words bidirectionally. "Bidirectionally" means predicting the word by analyzing the context before and after the word.

- Next Sentence Prediction:
  During this process BERT gets two or more sentences as input, and then it predicts the sequence of these sentences.

- Training Phase 2: Fine Tuning BERT to make it perform specific tasks. For example, let's consider that BERT answers the questions we input. We can achieve the results by replacing the fully connected output layers of the network with a fresh set of output layers that can output the answers to the

questions we want. Then we can perform supervised training using a fresh question and answer dataset.

Perplexity:

- The best language model is the one that predicts better on an unseen dataset.
- Perplexity is the inverse probability of the test set.
- Minimizing perplexity is the same as the maximizing the probability.

Entropy:

- It is defined as the log of the product where product is the product of probabilities of a event taking place.
- In theory, entropy of a random variable is the average level of “information”, “surprise”, or “uncertainty” inherent in the variable’s possible outcomes.

Evaluations:

How good should be a model?

- Assigning higher probability to “real” or “frequently observed” sentences rather than “ungrammatical” or “rarely observed” ones.
- Training parameters of the model on a training set which is never seen before by the model.
- An evaluation metric tells us how well our model performs on a training set.

Findings:

- The more context you will provide to the model, the more likely it is able to predict what comes next in the sequence.
- For example, unigram model can find out the next probable word even if the context provided is less.