Plan for week 4

1. Preparing for initial presentation: Draft before next meeting and final version before the end of next week

   The slide is nearly completed and shall be able to hand in before the meeting

2. Finishing literature review: complete 'n-gram' session and modify other part based on the information gathered and the draft

n-gram

N-gram: a language independent approach to IR and NLP

N-grams are sequences of characters or words extracted from text. N-gram can be divided into two categories: 1) character based and 2) word based. Character n-gram is a set of N consecutive characters extracted from a word. The main motivation behind this approach is that similar words will have a high proportion of common n-grams. The typical value of n is 2 or 3; they correspond to the use of bigram or trigram, respectively. Character based n-gram is usually used to measure the similarity of strings. Spelling, stemming and OCR are some applications that use character based n-grams. The word n-grams is a sequence of N consecutive words extracted from the text. Word level n-gram model has strong robustness for language statistical modeling and information retrieval, and it does not depend on language very much.

N-gram in language modeling

A language is modeled by using language and common sense knowledge. Formally, the language model is the probability distribution of word sequences or n-grams. Specifically, the language model estimates the probability of the next word given the preceding word. The word n-gram language model uses the history of the immediately preceding n-1 words to calculate the occurrence probability p of the current word. The value of n is usually limited to 2 (binary model) or 3 (ternary model). If the vocabulary size is m words, then in order to provide a complete coverage of all possible n word sequences, the language model needs to be composed of M^n-grams. Generally, n-gram language model only lists the most frequent word pairs, and uses backoff mechanism to calculate the probability of not finding the required word pairs.

3. Start coding part: carry out a plan and initial structure of the project.
   In order to make improvement on the lexicon-based classifier, we need to learn a pure lexicon-based classifier. In order to study how a lexicon based estimate the outcome, VaderSentiment which has a lexicon built manually with more than 7000 words, can be a good example. Each word is scored by 10 people, ranging from - 4 to 4. The average of these scores is the weight. Given an input text, the classifier will give a negative, medium, positive trend result and a composite score. Thus, the study of codes into vaderSentiment should be continuing processing.
   Then after literature learning I need to try to implement some simple n-gram coding, and considering how to apply it on vader, replacing unigrams.

4. Poem processing
   I have little progress on it and may continue searching for papers in spare time