CS 4740 Project Two Proposal

Xinyun Tang (xt222), Zhuofan Li (zl329), Zhoutong Li (zl683)
Kaggle TeamName: TangTang

- **Key points for HMM**
  The key point of building an HMM model for NER tagging is to match all the components in general HMM with specific part in NER tagging. The parameters are:
  Hidden variables: BIO NER tags
  O: Observed variables: word tokens
  A: transition probabilities: P(BIOTag_i | BIOTag_i-1) for a bigram model
  B: emission probabilities, or lexical generation probabilities: P(w_i|BIOTag_i)
  States: BIO NER tags
  π: initial probability distribution over BIO NER tags.

- **Pros and Cons about HMM**
  The pros are that HMM has reasonable performance for NER, while as a generative model, its drawback is that it cannot make use of the various features. HMM also requires efforts in dealing with unknown words.

- **Differences between HMM and MEMM**
  HMM is a generative model (generates the input), and MEMM is a discriminative model (conditions on the features from the input). MEMM compute directly the tags conditioning on the observation words and other features, while HMM computes the likelihood of observation word given the tags. MEMM is easier to incorporate a lot of, including long distance, features while HMM cannot.

- **Features for MEMM model**
  Possible features: preceding tokens, their corresponding p-o-s tags and BIO NER tags, current token (the observation word itself) and its p-o-s tag, following tokens and their corresponding p-o-s tags,

- **Baseline model description and performance**
  We applied a simplistic maximum likelihood estimation for each word in the training data to choose the most frequent tag for each word. As for the unknown word in the test data, we just randomly choose any tag. Since there is no parameters to optimize in this approach, we didn't use a validation set and used the entire training text to build the word tag dictionary.
  With this approach, we were able to get an accuracy of 52.33% on kaggle.