3) For POS tagging, we first consider the word/token itself. Looking at the word, we might have a general idea of what it can be tagged with. The next thing is to check the nearby POS tags. For example, if the token to the left is an adjective, then the current token can’t be a verb. This gives contextual clues that generalize beyond individual words.

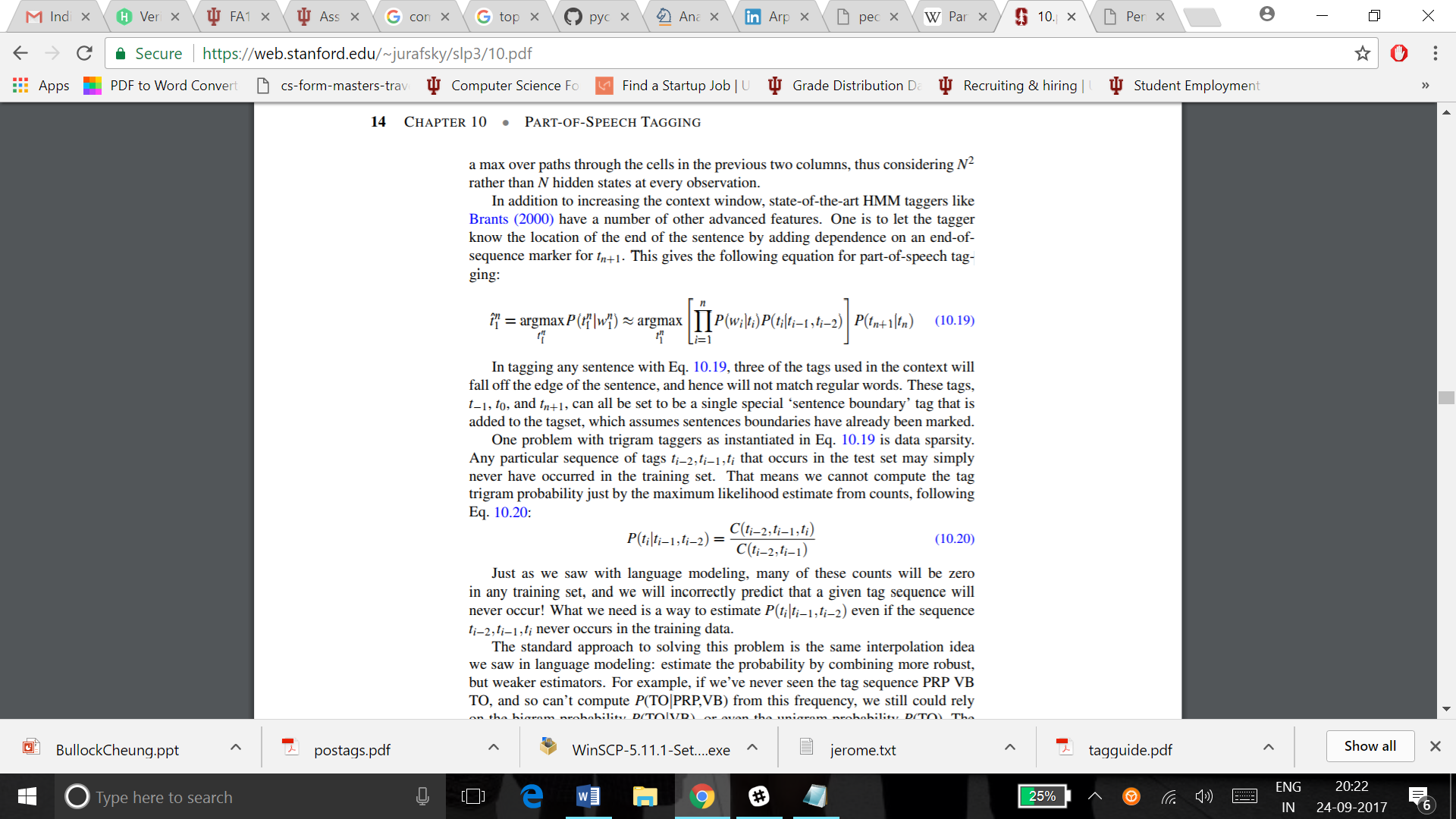
The Hidden Markov Model considers the two points above and helps with predictive tagging. However, to determine the best tag for a token, the decisions about some tags might influence decisions for others.

When there is sufficient training data, we are more likely to use trigram models. For a trigram tagger, to tag a current token, it considers the previous two tokens to get a better idea of the context. This improves the accuracy. The Trigram tagger assigns the part of speech tag correctly about 96% to 97% of the time. The accuracy can be expected to improve as the training lexicon grows. Trigram taggers general select the tag with high probability. This is how It works:

P(T, W) = P(t1, . . . , tn, w1, . . . , wn)

= P(t1|S)P(w1|t1)P(t2|t1)P(w2|t2). . . P(tn|tn−1)P(wn|tn)P(E|tn)

Under HMM tagging, there is basic assumption about trigram models, i.e. the probability of a tag is dependent on its surrounding tags. Using advanced features like letting the tagger know the location of the end of the sentence by adding dependence on an end-of sequence marker for tn+1 is one good method. Considering the prediction of tag ‘tn’, the three tags used in the context will be t−1, t0, and tn+1. This assumes sentences boundaries have already been marked. This gives the following equation for part-of-speech tagging:



The Viterbi algorithm comes in handy. It is the process of discovering the sequence of hidden states, given the sequence of all observations. This can be used with the trigram tagger.

In the example given, “I can help”, the token ‘can’ is tagged based on the previous two tags. i.e. the tag of the word ‘I’ and the start of the sentence, <S>. Here we are predicting the tags and not the words. Now the tag of ‘can’ can be right or wrong based on the tags surrounding it. In the above sentence ‘can’ is tagged as MD which is a modal.

For the sentence, “I can tuna”, can should have been tagged MD as well based on the previous two tags. But it is generally tagged as VBP. This is because of the word “Tuna”. The trigram tagger considers the words surrounding the token whose tag is to be found. Considering the surrounding tokens leads to a better prediction of the of the current token and hence the accuracy increases.