Name: Daniel Ribeiro Silva
AndrewId: drsilva

1. Usually, there is a trade-off between using less memory and increasing processing time. But if the goal is only to reduce RAM usage, without worring about processing time, we could do things like:
    - Storing numbers takes a lot less space than strings. So we could associate a numeric ID to each token. The higher the number of labels you have, the better the efficiency of this action, because before you had one dictionary of strings per label, and now you only need 1 string dictionary (token->tokenId) and all other will be (tokenId->count). Such an action would drastically reduce the amount of required memory.
    - For training, one approach would be to do what we saw in class last lecture: instead of keeping a dictionary in memory, just write to disk one line for each toke you see in a structured way, then sort and combine the results when needed.
    - For testing, a more drastic (very time-inefficient) solution would be to write the dictionary of each label to the disk and only load to memory the dictionary you are currently using (or, even more drastically, just read the entry you need from the disk).
    - A mid-ground solution for the above proposition would be to keep in memory only the dictionary for the frequent words and store to the disk the count for rare words. That way, most of the time would have in memory the count you need (the one for frequent words) and would only occasionally need to look for some count on the disk.
2. One way to do it would be to compute the log-likelihood for all labels and, for a given test example, predict all labels that have a log-likelihood higher than a given threshold t, or just the top-K (probably k=1) if no labels are above that threshold.
3. The importance of smoothing is to better represent statistically the distribution of the features (tokens) in a context. If we don't use smoothing, rare features may have a very distorted distribution. It could also happen that one rare feature that appears during testing could have had 0 occurrences during training. That would mean that the likelihood of that probability is 0 and, as consequence, the log-likelihood would be $-\infty$. That would crash the model. One reason to use Laplace smoothing is that it eliminates all those bad effects of rare features, since it will always impose a probability that is a middle ground between the empirical estimator and a uniform distribution.
4. Laplace smoothing corresponds to the expected value of the posterior distribution if we use the Dirichlet distribution with parameter α (the numerator contribution on smoothing) as a prior [1].
5. No / No

[1] http://en.wikipedia.org/wiki/Additive_smoothing