Homework #6

1. (30 pts) Consider again Peter Norvig's spelling corrector.

   Try to improve it by focusing on its shortcomings, and change **either** the language or the error model, e.g. you can use your knowledge of language models to improve upon the unigram version. For example, can you make the corrector change "cae" in "car" after "fast", but "cat" after "furry"? Or, you can use your knowledge of the English language and keyboards to produce a more realistic error model.

   Change the code and motivate in English your choice, explaining what you are trying to achieve (the actual quantitative improvement matters less than your code and explanation!).

2. (30 pt) Consider the text classification notebook we discussed in class.

   Part 1

   Refactor the text classification code from a notebook version to a flow version, i.e. porting the steps we have discussed into Metaflow: one step will load the data, one step will pre-process it etc..

   Don't forget to track the results of model evaluation in Comet, and some qualitative tests before ending the flow, to make sure the model is not making "dumb predictions". Heavily comment your code to explain why you pick a certain test case (i.e. show your work conceptually, not just as working code).

   Part 2

   Re-use the code from Part 1 to investigate the role of different pre-processing techniques in our classification task. In particular, take advantage of Metaflow branching option to evaluate 3 different text pipelines in parallel, and then use a JOIN step to pick the best one according to some quantitative criterion.

   Before ending the flow, remember to log your results in Comet and to add some qualitative tests to verify that the model you chose based on a quantitative metric is indeed performing as expected.