I'm going to suggest groups of 2–3 people to work on this assignment. It's not quite as involved as the Stanford CoreNLP one, but still has some technical start-up time and much thought involved.

1. (a) We didn't talk about Kneser-Ney smoothing in class, but I want you to read up on it (finding your own sources) and summarize how it works in half-a-page or less.

(b) Download and install KenLM (https://kheafield.com/code/kenlm/), a language modeling toolkit which employs Kneser-Ney smoothing.

• Note that you need to first install Boost (http://www.boost.org). See the Getting Started guide for more information: http://www.boost.org/doc/libs/1_59_0/more/getting_started/ ... e.g., following the KenLM advice, my boost-installing command (after running the bootstrap.sh code) was:

   $ sudo ./b2 --prefix=/opt/local/ --libdir=/opt/local/lib/ --layout=tagged link=static,shared threading=multi,single install -j4

• Note: to install KenLM, I had to run the bjam -j4 command with the --with-boost flag set to the proper directory, e.g.:

   $ ./bjam -j4 --with-boost=/opt/local/

(c) Train the LM on some collection of text, choosing files at your discretion. (See the documentation on Estimation for training.)

(d) Take the holbrook-tagged corpus of misspellings here: http://www.dcs.bbk.ac.uk/~ROGER/corpora.html. You are going to use your LM to determine a good correction for every misspelling marked by an <ERR> tag. A few notes on the task:

• Use your LM to compare the probabilities of the original word with one or more spelling correction candidates (See the documentation on Querying.). Use the same training data you used to train KenLM to obtain correction candidates—though, how you obtain your candidates is up to you. (Hints: Levenshtein distance, SOUNDEX, confusion sets, etc.)

• We are simplifying the task by assuming that error detection has been done for us; we are simply correcting. (You may extend your system to do detection as well, but make sure you have the correction module in place first.)

• (You are not allowed to use the gold standard correction of one word to feed into the LM for surrounding errors; you may, however, use a guess from your system, if you desire.)

(e) Report your work and your resulting accuracy in the corpus. If you have time, experiment with class-based models and different OOV handling (see again the documentation on Estimation). Do you get different results?