Spell Correct

By : Abhinav Rai, rai.1@iitj.ac.in

Recommed reading my blog on Language Modelling here and on Spell correction theory here.

This has majorly two parts – 1. Edit Model 2. Language Modelling

Data

We will be using the writings of secondary-school children, collected by David Holbrook. The training

data is located in the data/ directory. A summary of the contents:

• holbrook-tagged-train.dat: the corpus to train your language models

• holbrook-tagged-dev.dat: a corpus of spelling errors for development & test.

• count_1edit.txt : a table listing counts of edits x|w, taken from Wikipedia.

More information on Holbrook Data - The material is presented here in two forms. The file *holbrook-tagged* contains the passages pretty much as published except that the misspellings are tagged. For example <ERR targ=sister> siter </ERR> means that "siter" was written for "sister". In a few cases, where I could not decide what word was intended, a question mark is given where the target ought to be.

We make training corpus and dev corpus. In this format. - “His sister (siter) is there.” and add <s> & </s>

Language Models used -

• **Laplace Unigram Language Model**: a unigram model with add-one smoothing. Treat out-of-vocabulary items as a word which was seen zero times in training.

• **Laplace Bigram Language Model**: a bigram model with add-one smoothing.

• **Stupid Backoff Language Model:** use an unsmoothed bigram model combined with backoff to an add-one smoothed unigram model. In this, I have made simple backoff with Trigram which was giving better accuracy than Bigram model**.**

• **Custom Language Model**: Trigrams and interpolation. Best accuracy in all 4.

Implementing Language Model – Functions

• train(HolbrookCorpus corpus): takes a corpus and trains your language model. Compute any counts or other corpus statistics in this function.

• score(List words): takes a list of strings as argument and returns the numerical score, which should be the log-probability of the sentence using your language model.

Files and Functions

Datum.py

- Datum class contains word and error. Any datum may be with error or not. Eg. Object d of Datum class can have d.word = sister & d.error = siter.
- Also contains simple functions like fixError, hasError & isValidTest.
- IsValidTest uses dameraulevenshtein function implemented in EditModel.py.

Sentence.py

- Class made to define sentences. It is constituted of list of datums.
- Simple functions are implemented in sentence class to support sentence operations like get, put, getErrorSentence, etc.

HolbrookCorpus.py

- Class where the actual data is removed from <err> tags and put in datum, sentences and corpus.

EditModel.py

- Here the data in Confusion Matrix is read and made into edit table which is useful in computing probabilities and return the values.
- Additional comments have been made and examples are also shown as how they are implemented in the file.

SpellCorrect.py

- contains main | imports all other files | Master file in the project.
- Loads training corpus and development corpus (testing).
- Firstly the model is made. We train the model with the training corpus.
- Now the edit model is made with this language model in SpellCorrect class.
- Then the model is evaluated with many sentences (having only 1 correction each)

Acknowledgement: This is adapted from Coursera assignment by Christopher Manning & Dan Jurafsky.

Results (Accuracy):

•Laplace Unigram Language Model: 0.11

•Laplace Bigram Language Model: 0.13

•Stupid Backoff Language Model: 0.18

•Custom Language Model: 0.19

The performance we expect from your language model is not that great! We have provided with a very simple edit model, not a lot of training data. This is great accuracy with this small data set. With large data set, the algorithm remains same and accuracy increases.

For any further details/suggestions feel free to contact the author and owner of this repo.