Tomer Zwi
Tz2247

Project Proposal: A Simple Suggest Word

I would like to create a tool in Haskell that suggests words to complete sentences based off of the user's input. The idea comes from my interest in NLP as well as my daily use of "auto suggest tools" over text and email. I think creating this tool in Haskell would be interesting.

The project has two parts:
Part 1 is collecting, cleaning, and storing a dataset, which was obtained from thousands of Yelp.com reviews. For this, I will run in parallel and not in parallel and compare between them.
Part 2 is testing the program and getting user input (incomplete sentences that need completing) from the command line and checking if there are any suggested word. This will return a result or error message.

Following is more details about each part discussed above:

Part 1:

-> Collection - My dataset will come from a Yelp reviews dataset (link below). I will have 20 documents or less that will be in similar size/amount of reviews and I will run the program on them. The reason I choose user reviews is because the language used is more casual and conversational than the language in books.

->Preprocessing and cleaning the dataset - the program reads the data line by line and does the preprocessing/cleaning by

# Removing stop words from the sentence such as "the", "a", "at", "on" and etc.
# Making all words lowercase
# basic stemming: removing "s", "ed","ing" at the end of the word. To do this, I found a library in Haskell called NLP. snowball that I might use (if it doesn't work, I will try to find rules to do it manually)
# remove punctuation
#remove words that have digits within them

->Storing the data - after cleaning, I will have a list of words. I will create two tables: bigram (sets of two words) and trigram (sets of three words).
The bigram table will show for each two words {"bigram": number of occurrences: probability of occurerence}
The trigram table will show for each three words {"trigram": number of occurrences: probabilities}.
The field of "probabilities" will help me to rank the potential words that will be suggested by the program.

To store the data I am planning to use “sqlite”:

https://hackage.haskell.org/package/sqlite-simple

Part Two:

The user inputs on the command line either one or two words. There are three output scenarios:

1. Error number 1 - Error because it could not find: "there is no bigram or trigram that contains the words." This would also occur if they input a non-english word.
2. Error number 2 - Error because the user input an invalid input such as 0 words or more than two words.
3. Suggestions words - the user’s input is valid and there are suggestions from the database. The program will return up to 5 suggested words which are ranked by the highest probabilities.

Source for Yelp dataset:
https://www.yelp.com/dataset