TextTippyLite: An app for word prediction

Jennifer Holtzman, Ph.D.
May 9, 2018

The SwiftKey Dataset

For the Data Science Specialization final project, students were supplied with a large collection of English text from web sources of three types, news, blogs, and twitter. The aim was to use this data to develop an app in Shiny that would predict the next word in a phrase entered by a user.

The word prediction program was developed in R using the 'quanteda' package, which contains several efficient tools for natural language processing (NLP).

Two search problems considered

Language is governed by patterns and grammar rules, but it is also infinitely creative. The corpus of text data (90 % sample from the combined web sources) was processed in different ways to build databases that present features along with their frequencies. These are used to solve the types of prediction problem inferred from quiz questions.

- NLP problem 1: extending common phrases or word groupings
- NLP problem 2: prediction based on keywords to extract context (developed but not deployed due to memory constraints)

Database strategies

N-gram databases were prepared by first reading each document in the corpus with a sliding window of fixed word size (tokenizing), counting the frequency of the observed features (words or word groups), and ordering by descending frequency. TextTippyLite only uses an N4 database, which is a frequency table of four-word phrases.

Tokenized text was cleaned by removing punctuation, numbers, and extra whitespace. Database size was reduced by including only words from the English dictionary and removing low frequency N4-grams, i.e., features with frequency < 5.

Queries and searches

Input phrases are parsed into queries before searching the databases. Searches are executed by constructing regular expressions and using the 'grep' function. Use of spacing reduced unintended results (e.g. query " here " will not return "there", "heretic", etc.).

To complete common phrases, the last 2 or 3 words of the input were used to construct the query. The query is then was used to search against an N4 database.

The app

https://fire-elf.shinyapps.io/texttippylite/

The app pre-loads the N4-gram database.

User inputs a phrase into a textbox.

Radio buttons allow the user to choose between the search types (placeholder for future alternate search type).

The top results are presented in a table.

Possible future exploration and development

- Context-based search using quanteda's feature colocation matrix (sparse matrix) paired with bigram database (developed but not deployed)
- Word predition for rare / obscure phrases
- Using hash coding, caching, or SQL to increase speed / efficiency of database searches
- Improved handling of proper nouns and associations (e.g., names of well-known people associated with field; Wikipedia might be a useful starting point)
- Semantic tagging of the corpus to help with grammatical prediction
- Handling of synonyms