Capstone: Milestone Report

Tim Sperzel
January 24, 2020

Introduction

This project analyses the HC Corpora Dataset with the end goal of creating a Shiny App for predicting n-grams. This first milestone report summarizes an exploratory data analysis.

File Summary

Three data files sourced from blogs, news, and twitter were read into R. The news file had hidden null characters preventing a full file read and these null characters required hand deletion with Notepad++ prior to file loading.

f_names f_size f_lines n_char n_words pct_n_char pct_lines pct_words

---

blogs 200.4242 899288 208361438 37334131 0.36 0.21 0.37 news 196.2775 1010242 203791400 34372528 0.35 0.24 0.34 twitter 159.3641 2360148 162385035 30373583 0.28 0.55 0.30

Processing files of this size pushed up against R's memory limits and ran slowly. To facilitate analysis, we sampled ten percent of the lines from each file. We cleaned the sample and created n-grams. To further speed processing, we subsetted the n-grams to those that covered 90% of the sample phrases.

Uni-grams

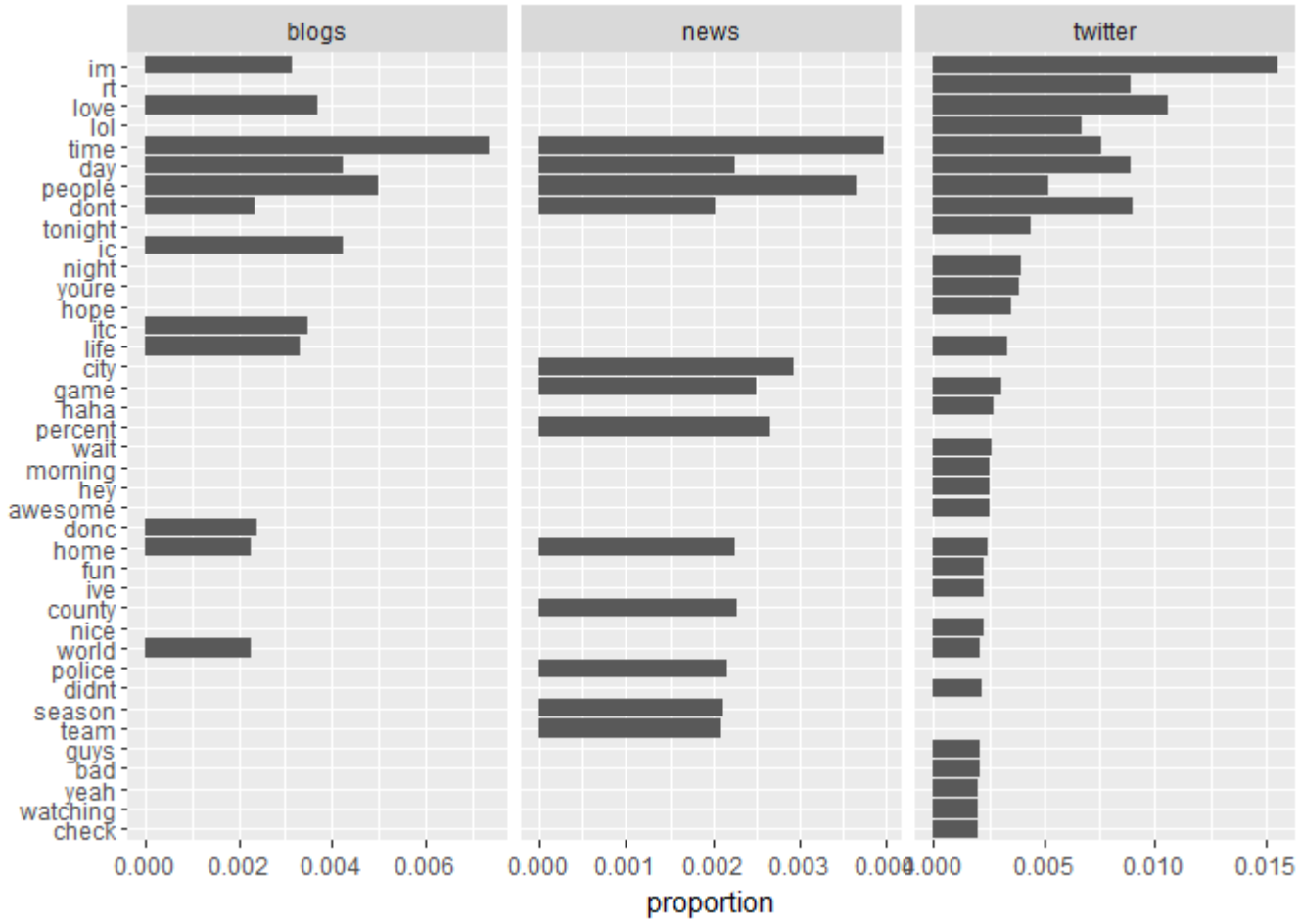
The corpora are populated with many acronyms and abbreviations such as "rt" for re-tweet, "lol" for laugh out loud, "ic" for I see. Notably, we chose to leave the short hand "im" for I am and "dont" for don't / do not as is, hence they show up as uni-grams.

Uni-gram Wordcloud

Word distribution can be summarized with a word cloud, where word size/color represents frequency. The words, "im", and "time" show up as most frequent followed by "people", "dont", "day", and "love". This is a popular visual method, but we prefer the relative frequency column plots shown below.

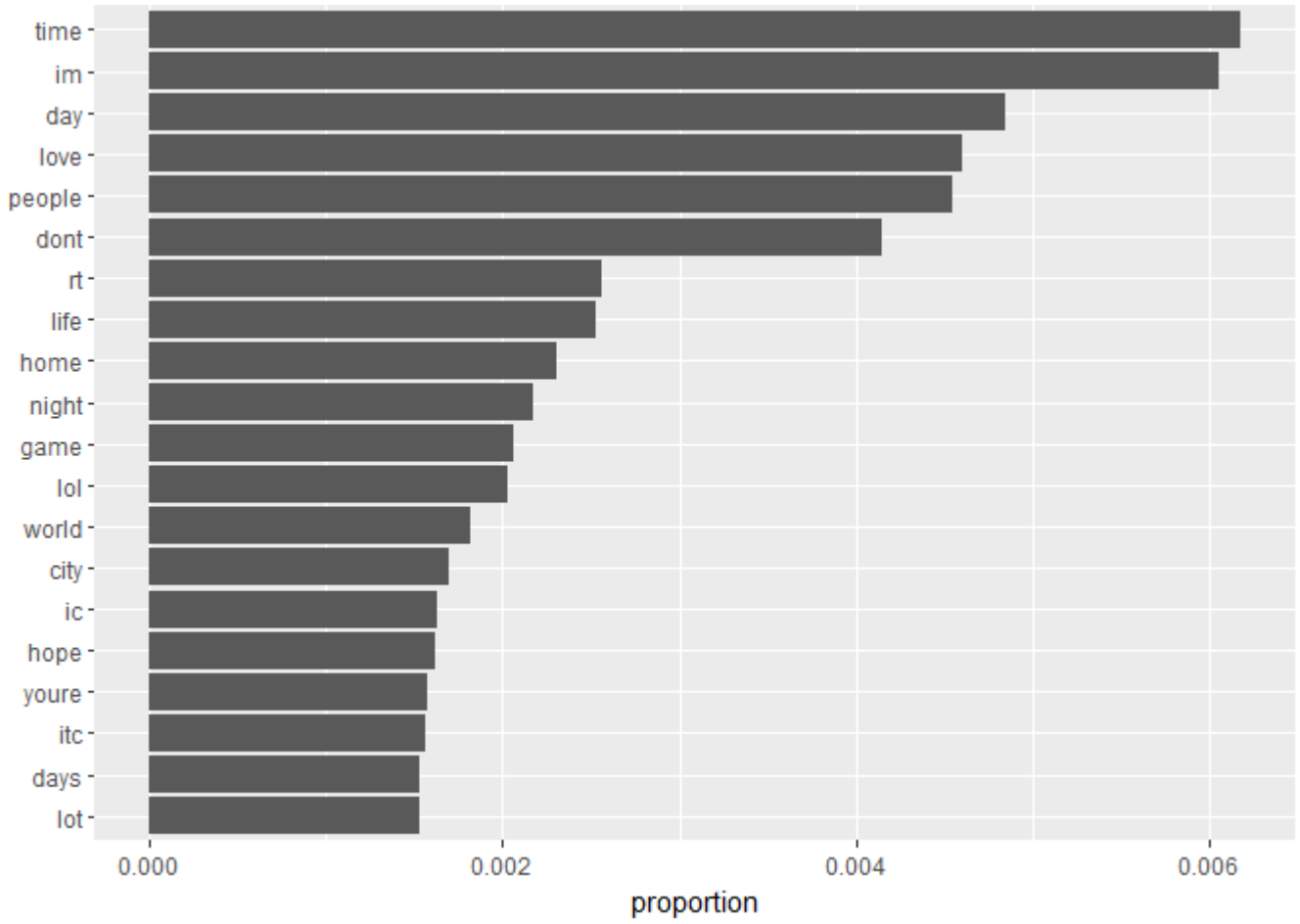
Uni-grms, By Source

The different files - blogs, news, twitter - had different word relative frequencies. Notice that in terms of most frequent words, "rt" occurs only on twitter, "ic" and "donc" only in blogs, and "city", "percent", "county" only in news.

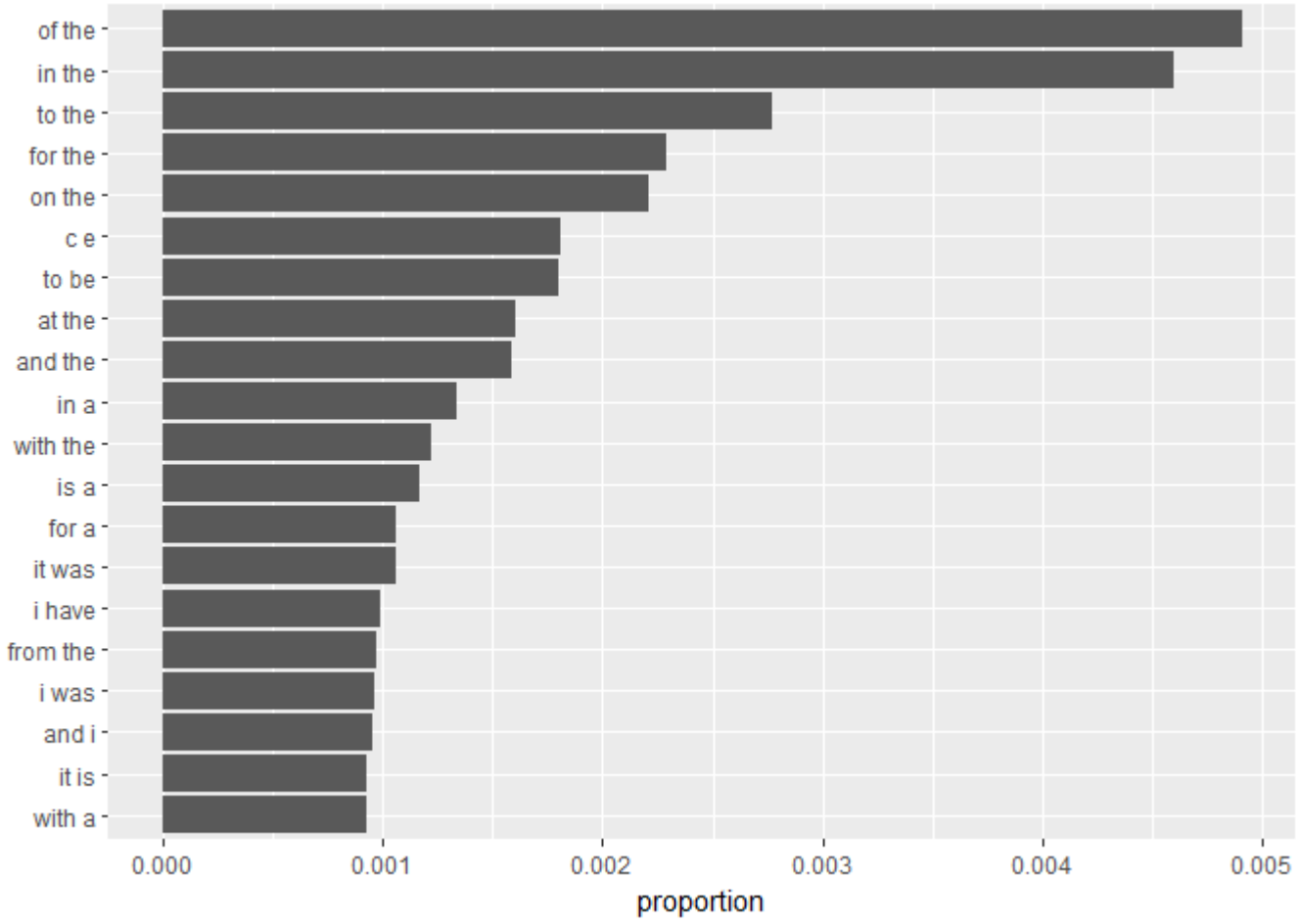


Uni-gram Distribution

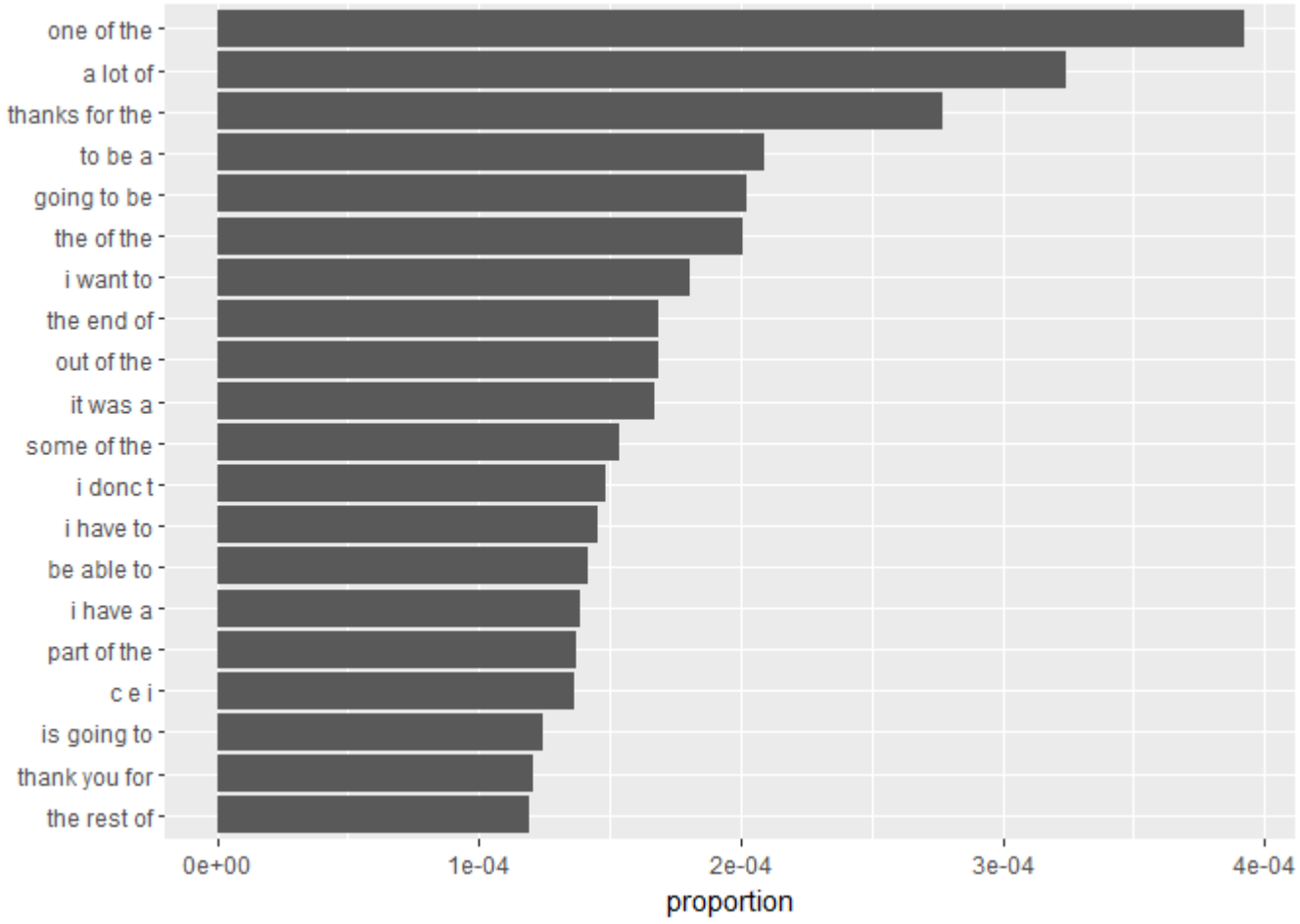
Distributions were created for each set of n-grams, based on relative frequency.



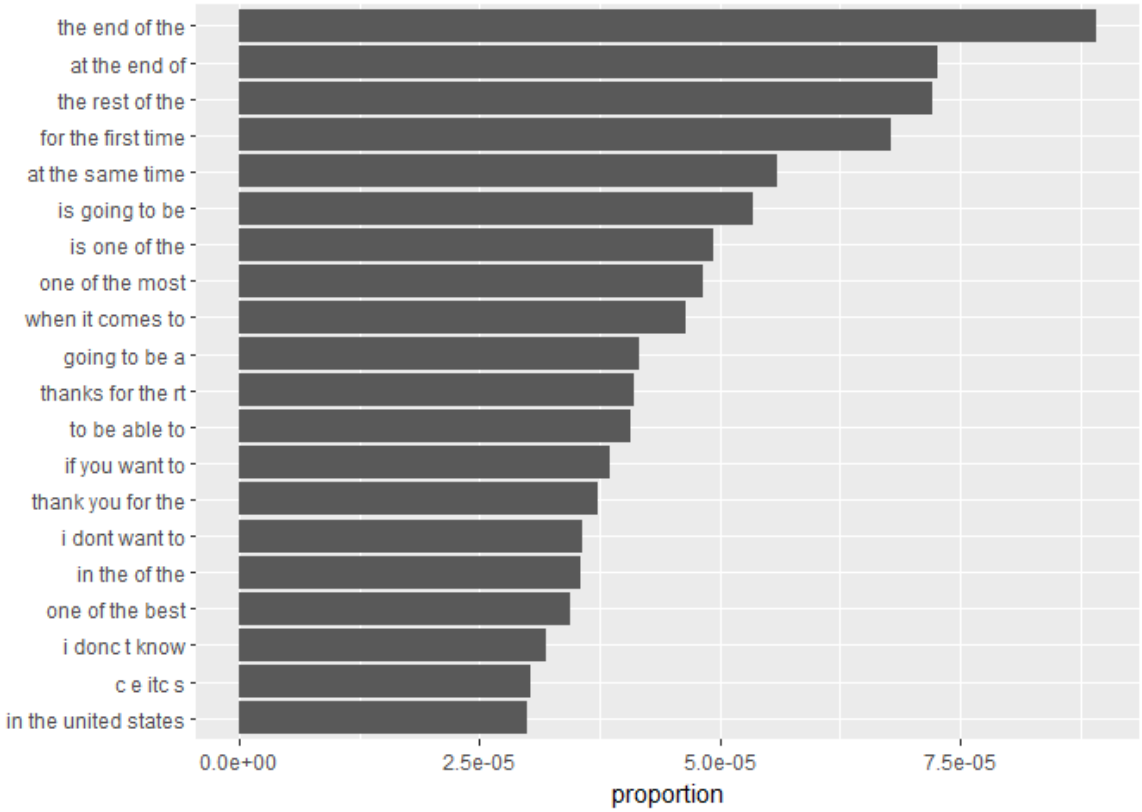
Bi-gram Distribution



Tri-gram Distribution



Quad-gram Distribution



N-gram Prediction Model

I anticipate using the n-gram tables created for bi-gram, tri-grams, and quad-grams as the basis for prediction. The user will input a word, the model will find the bi-gram with the greatest relative frequency given that word. Similarly, the tri-gram table will be used for making predictions from two word entries and so on.

word1 word2 word3 word4 n proportion coverage

---

the end of the 806 8.93e-05 0.0000893 at the end of 656 7.27e-05 0.0001619 the rest of the 651 7.21e-05 0.0002340 for the first time 613 6.79e-05 0.0003019 at the same time 506 5.60e-05 0.0003580 is going to be 482 5.34e-05 0.0004113

Notice in the guad-gram table, that the 4-grams are separated by word and arranged by relative frequency. When the user inputs three words, the model matches those words and then finds the fourth word with the greatest relative frequency. Cases where there is no match, or where more than three words are entered, will have random completion.