Text Reuse in Large Corpora

Johnathan Edwards

5/4/2016

The Project Overview

Plagiarism is a major concern in academia. In recent years there has been an overall increase in the number of cases of plagiarism in universities across the country. This is due to an increase in the effectiveness of detection capabilities. I am interested in this topic because it can have an impact on research in any field.

The Data

The Webis Crowd Paraphrase Corpus 2011 (Webis-CPC-11) contains 7,859 candidate paraphrases obtained from Mechanical Turk crowdsourcing. The corpus is made up of 4,067 accepted paraphrases, 3,792 rejected non-paraphrases, and the original texts. These samples have formed part of PAN 2010 international plagiarism detection competition, but were not previously available separate to rest of the competition data.

My Program

My program does not look for paraphrasing, instead, I am looking for plagiarism of any kind. Therefore, my result vary greatly from the meta data provided. However, using a simple sample of 50 documents, I verified a portion of the findings by hand.

What did not work

Working with such a large dataset proved to be very difficult using the NLP package in R. Scaling from a small document term matrix (two documents) for testing to a large document term matrix (greater than 7 , sadly) caused serious time issues despite using multicore methods. Even using an Amazon instance for greater than 100 documents proved too time expensive.

What did not work

The parallel package, and in fact, all multi-core packages in R do not work with the Windows operating system. Tring to multi-core on my laptop was not fruitful because my laptop uses a mobile processor.

How I addressed it

I created an 8 processor m4.2xl AWS instance of RStudio server. I then snapshotted the storage (10g) and used that to create and attach a new larger storage drive. This allowed me to upload my entire corpus and otehr test files onto the drive through SSH. This reduced my DTM creation time from hours to minutes. Additonally, I changed direction from traditional NLP packages to the TestReuse package. This package allowed me to parallelize my corpus creation and provided methods for me to analyze my data.

What did not work

I approached this data set incorrectly. I approached looking for document similarity in an attempt to find plagiarism the issue is that I could not get my meta data for the data set to properly verify my findings. The issue was that the meta data said if there was paraphrasing, not copying. Therefore, I found plagarism that was not necessarily paraphrasing because it may just be a direct quote. The meta data did not account for this.

How I addressed it

I, really a friend and I, verified by sight 100 documents 50 with high scores that indicate documents of high similarity and 50 with low similarity scores. This allowed me to see how my meta data was incorrect.

Create the minihash and corpus

```
setwd("~/Dropbox/corpus-webis-cpc-11/")
minhash <- minhash_generator(200, seed = 235)
corpus <- TextReuseCorpus(dir = "Webis-CPC-11",
                          tokenizer = tokenize_ngrams, n = 5,
                           minhash_func = minhash)
```

Corpus Creation Result

```
## TextReuseCorpus
## Number of documents: 6
## hash_func : hash_string
## minhash_func : minhash
## tokenizer : tokenize_ngrams
```

Find Candidates

```
 buckets <- lsh(corpus, bands = 50, progress = TRUE)
candidates <- lsh_candidates(buckets)
```

```
head(candidates)
```

```
## Source: local data frame [6 x 3]
##
##                  a               b score
##              (chr)           (chr) (dbl)
## 1   1004-original 1004-paraphrase    NA
## 2   1012-original 1012-paraphrase    NA
## 3   1012-original    552-original    NA
## 4 1012-paraphrase    552-original    NA
## 5   1015-original   1016-original    NA
## 6   1015-original   1017-original    NA
```

Calculating the Scores

```
scores <- lsh_compare(candidates, corpus,
                      jaccard_similarity, progress = TRUE)
```

```
head(scores)
```

```
## Source: local data frame [6 x 3]
##
##                  a               b     score
##              (chr)           (chr)     (dbl)
## 1   1004-original 1004-paraphrase 0.2068966
## 2   1012-original 1012-paraphrase 0.8510638
## 3   1012-original    552-original 0.5689655
## 4 1012-paraphrase    552-original 0.4827586
## 5   1015-original   1016-original 1.0000000
## 6   1015-original   1017-original 1.0000000
```

Cleaning the Scores

```
 cleanedScores <-scores %>%
      filter(!grepl("original", b)&!grepl("paraphrase", a)) %>%
      filter(!grepl("metadata", b)&!grepl("metadata", a)) %>%
      filter(sub("-.*","", a) == sub("-.*","", b))
  colnames(cleanedScores) <- c("Original", "Suspect", "Score Similarity" )
```

Final Scores

```
head(cleanedScores)
```

```
## Source: local data frame [6 x 4]
##
##        Original         Suspect Score Similarity    NA
##           (chr)           (chr)            (dbl) (chr)
## 1 1004-original 1004-paraphrase        0.2068966  1004
## 2 1012-original 1012-paraphrase        0.8510638  1012
## 3 1017-original 1017-paraphrase        0.7254902  1017
## 4 1026-original 1026-paraphrase        1.0000000  1026
## 5 1027-original 1027-paraphrase        1.0000000  1027
## 6 1058-original 1058-paraphrase        0.5316456  1058
```