Understand frequencies of words and word pairs

Alexander Alexandrov

Friday, May 06, 2016

Introduction

The first step in building a predictive model for text is understanding the distribution and relationship between the words, tokens, and phrases in the text. The goal of this task is to understand the basic relationships you observe in the data and prepare to build your first linguistic models.

Questions to consider:

1. Some words are more frequent than others - what are the distributions of word frequencies?
2. What are the frequencies of 2-grams and 3-grams in the dataset?
3. How many unique words do you need in a frequency sorted dictionary to cover 50% of all word instances in the language? 90%?
4. How do you evaluate how many of the words come from foreign languages?
5. Can you think of a way to increase the coverage – identifying words that may not be in the corpora or using a smaller number of words in the dictionary to cover the same number of phrases?

Clean the Data

Following should be cleaned:

1. punctation;
2. numbers;
3. stop wrods;
4. profanity words;
5. swear words;
6. URLs, emails, accounts.

```
## Loading required package: NLP
```

```
set.seed(123)
corpus <- ReadAndCleanCorpus("./data/en_US/", prob=0.001)
```

N-Gram's Frequencies

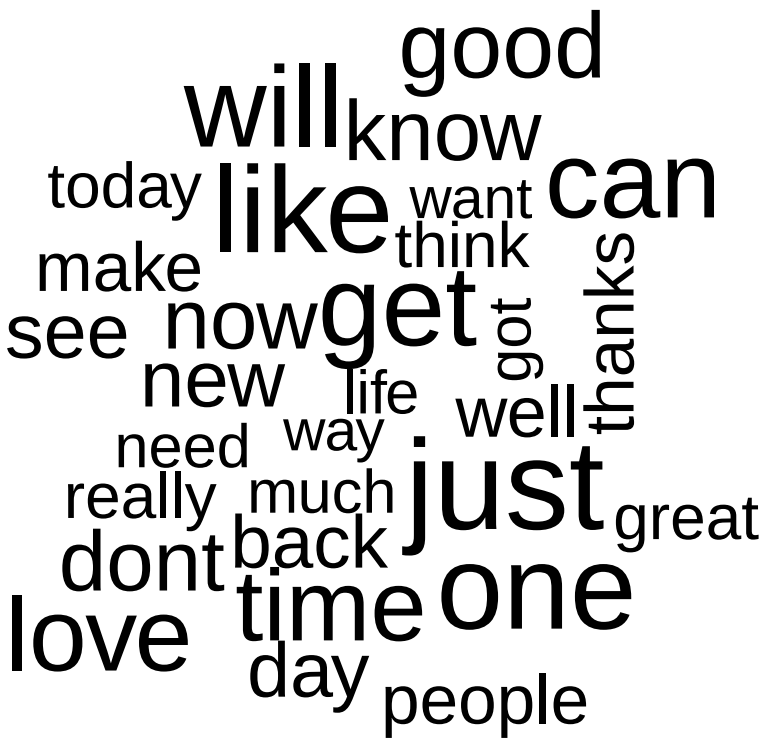
Compute Term's Frequencies

```
doc.term.matrix <- as.matrix(DocumentTermMatrix(corpus))
term.freq <- colSums(doc.term.matrix)
term.freq <- sort(term.freq, decreasing=T)
head(term.freq, 10)
```

```
## just like  one  get will  can love time good dont
##  240  229  220  211  209  202  188  184  168  151
```

```
## Loading required package: RColorBrewer
```

```
top.word.count <- 30
wordcloud(names(term.freq)[1:top.word.count], term.freq[1:top.word.count])
```



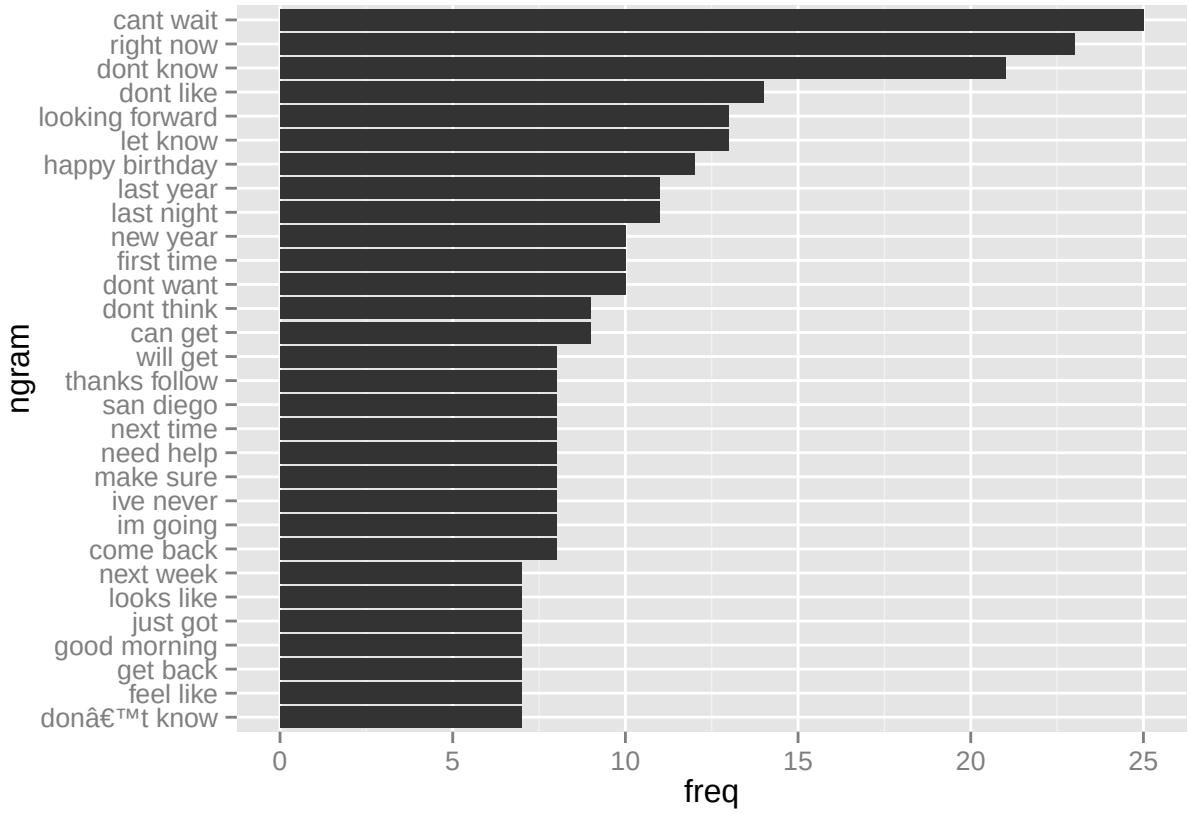
Compute 2-Gram's Frequencies

```
##
## Attaching package: 'ggplot2'
##
## The following object is masked from 'package:NLP':
##
##     annotate
```

```
getNGramsFreq <- function(corpus, n) {
    tokenizer <- function(x) NGramTokenizer(x, Weka_control(min=n, max=n))
    ngram.doc.matrix <- as.matrix(TermDocumentMatrix(corpus,
            control=list(tokenize=tokenizer)))
    ngram.freq <- rowSums(ngram.doc.matrix)
    return(sort(ngram.freq, decreasing=T))
}
drawTopNGrams <- function(ngram.freq, top) {
    top.ngram.freq <- data.frame(ngram=names(ngram.freq)[1:top],
                                 freq=ngram.freq[1:top])
```

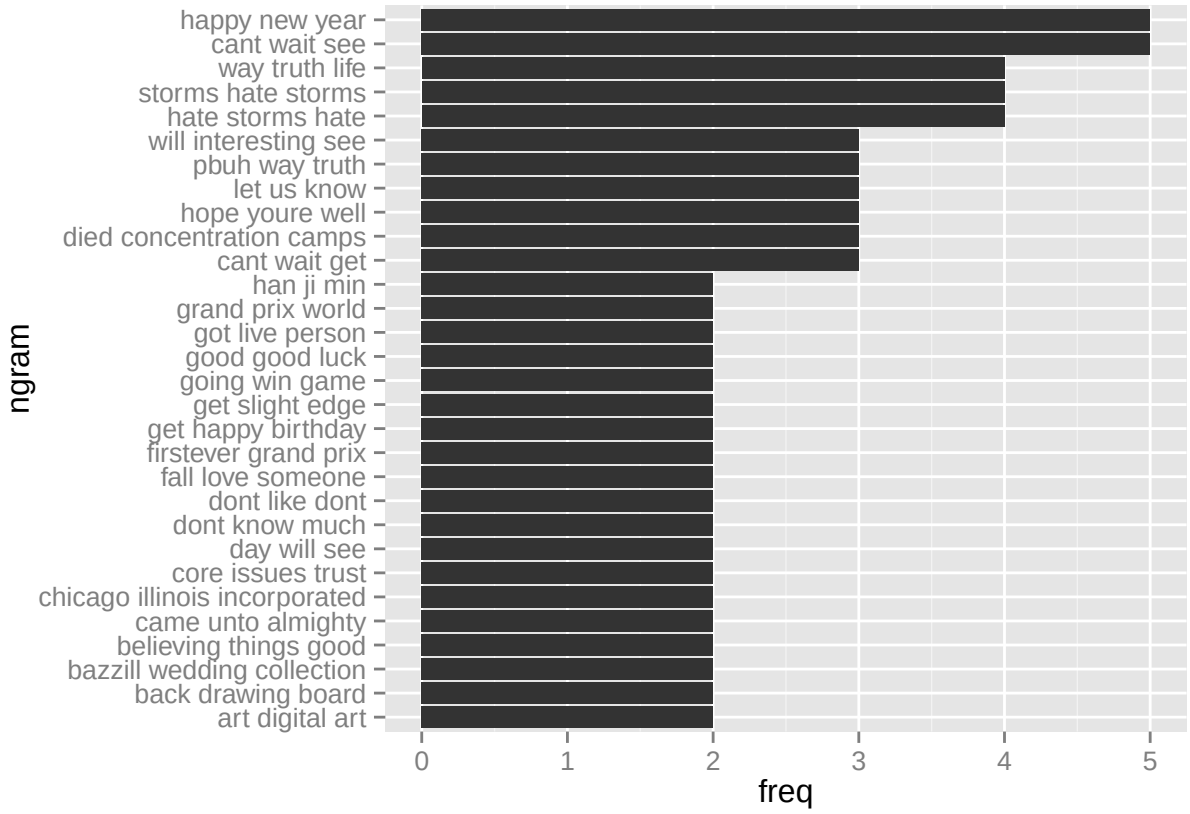
```
    top.ngram.freq$ngram <- reorder(top.ngram.freq$ngram, top.ngram.freq$freq)
    ggplot(top.ngram.freq, aes(x=ngram, y=freq)) + geom_bar(stat="identity") +
            coord_flip()
}
```

```
gram2.freq <- getNGramsFreq(corpus, n=2)
drawTopNGrams(gram2.freq, top=top.word.count)
```



Compute 3-Gram's Frequencies

```
gram2.freq <- getNGramsFreq(corpus, n=3)
drawTopNGrams(gram2.freq, top=top.word.count)
```



Dictionary reduction

To make better predictions dictionary shoud reduced and cleaned from very rare or occasional words and phrases. So, how many words covers 50% of all word instances in the language?