Ai Programming Evaluation

John Fee

March 6

Problem Statement

Using the given dataset for this evaluation, hereafter referred to as the "movies corpus", we answer the following open-ended questions.

1. What are the five most popular movie genres?
2. What words are characteristic of the movie genres from Question 1?
3. Does the distribution of word frequencies in the movies corpus follow Zipf's law?

Data Processing

The movies corpus consists of over 42 thousand unique movie records with information on the movie's title, release date, box office revenue, what genres the movie belongs to, and a summary of the movie's plot. No metadata was provided on the origin of the data or how it was collected, but we do note that it contains American and foreign (e.g. Bollywood) films.

To make analyzing the movie summaries tractable, we use Python to perform the following processing steps:

- Remove non-alphanumeric symbols to remove punctuation.
- Parse each summary into a sequence of tokens and lemmatize them (map similar tokens to the same word representation) using models provided by SpaCy.
- Join tokens which frequently co-occur into phrases using Gensim.
- Identify and remove "stop words", words which are too common to be useful for differentiating texts, by removing words which appear in over 50 percent of the summaries.

Table 1: Top five genres by revenue.

Genre	Revenue (trillions)
Drama	1.68
Comedy	1.58
Action	1.54
Thriller	1.37
Adventure	1.33

Popular Movie Genres

We define the popularity of a movie genre as the number of unique moviegoers who purchase tickets for a movie belonging to that genre. Given that ticket data is not available in the movies corpus, we use total box office revenue for all movies belong to the genre as a proxy variable and assume that this quantity is a monotonic (rank preserving) function of genre popularity. Furthermore, we assume that movies which are missing box office revenue (approximately 82 percent the records in the movies corpus) can be omitted from the genre popularity analysis without distorting the result. The resulting top five most popular genres is presented in Table 1.

Genre Keywords

We extract characteristic keywords from each popular genre by ranking tokens using the Term Frequency - Inverse Document Frequency (TF-IDF) metric. TF-IDF improves upon using word frequencies to rank word importance in a document by penalizing words which appear in many documents (which consequently are likely to be uninformative). For our analysis, each "document" of interest is the combined body of summaries associated with a single genre. The TF-IDF score is defined as

$$\text{TF-IDF Score(word,G,D)} = tf(word, G) log_2\big(\frac{D}{df(word}\big)$$

where $tf(word, G)$ is the term frequency (number of occurrences of the word) in the movie summaries associated with genre G, D is the number of genres in the movies corpus, and $df(word)$ is the document frequency (number of different genres the word appears in). The characteristic keywords for each genre are presented in Table 2. Note that there is significant crossover between some of the genres - the movie genres are not independent of each other!

Table 2: Characteristic words for top five most popular movie genres.

Drama

Word	tfidf_score
father	0.0846
marry	0.0722
love	0.0713
mother	0.0693
family	0.0692
tell	0.0690
get	0.0689
child	0.0657
son	0.0654
police	0.0620

Comedy

Word	tfidf_score
bugs	0.1344
tom	0.1041
jerry	0.1036
get	0.0867
daffy	0.0811
stooges	0.0780
sam	0.0654
car	0.0651
tell	0.0591
charlie	0.0589

Action

Word	tfidf_score
kill	0.1212
vijay	0.0881
police	0.0775
gang	0.0767
raja	0.0738
escape	0.0730
car	0.0690
shoot	0.0640
jack	0.0630
get	0.0624

Thriller

Word	tfidf_score
kill	0.1209
police	0.1079
murder	0.0947
car	0.0898
killer	0.0671
shoot	0.0669
david	0.0656
escape	0.0655
tell	0.0644
house	0.0630

Adventure

Word	tfidf_score
ship	0.1139
tarzan	0.0847
king	0.0844
earth	0.0807
sharpe	0.0781
kill	0.0725
planet	0.0716
island	0.0711
escape	0.0699
littlefoot	0.0684

Zipf's Law

Zipf's Law states that the distribution of word frequencies follows a power law distribution where the frequency of any given word is inversely related to its "rank" in the word frequency order. That is,

$$\text{word frequency}(word_k) \ \alpha \ \frac{1}{k}$$

where k is the index the of the kth most frequent word. If we explicitly model the word frequency as

$$\text{word frequency}(word_k) = c\frac{1}{k}$$

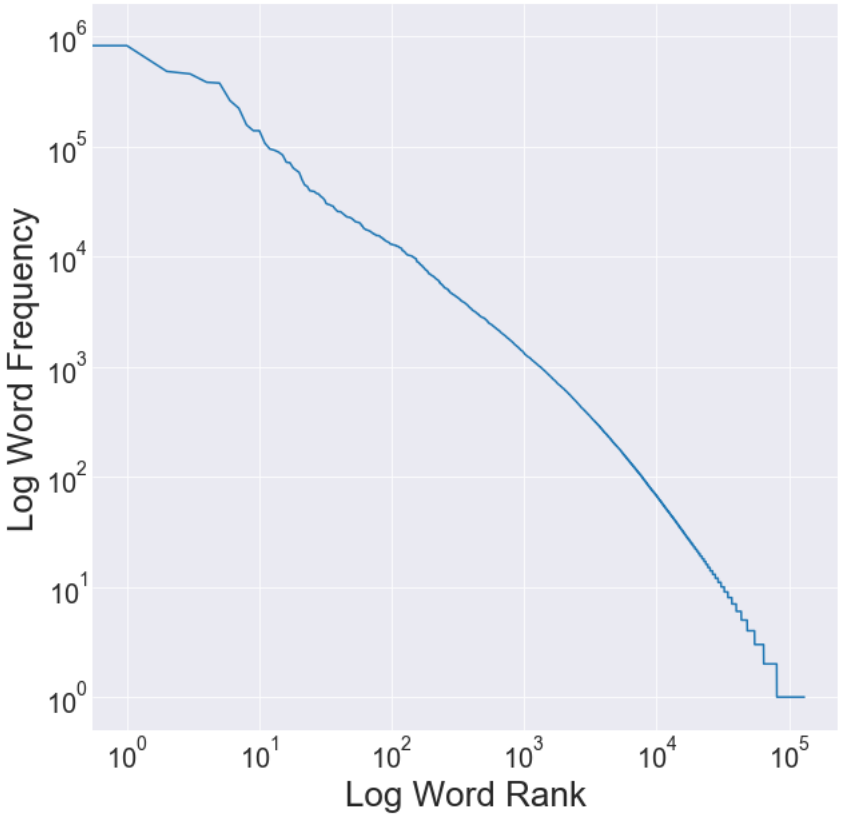
where c is a shape parameter, it follows that a log transformation of both sides of the equation

$$\begin{aligned} log(\text{word frequency}(word_k)) &= log(\frac{c}{k}) \\ &= log(c) - log(k) \end{aligned}$$

yields a negative linear relationship (between log(word frequency) and log(word index))with a slope of -1 which we can examine movies corpus for evidence of (or lack thereof).

As seen in figure 1, the log-log relationship is plausibly linear and demonstrates that Zipf's law is a good first approximation for word frequencies in the movies corpus.

Figure 1: Zipf's Law in action (stop words included).



Future Improvements

The TF-IDF method we used to extract keywords is a simple method that works well in practice, but there are (more sophisticated) alternatives. The Rapid Automatic Keyword Extraction (RAKE) algorithm, which works by extracting frequently co-occurring words (similar to how we handled phrase detection), shows promise. Treating keywords as a topic model problem, where the distribution of keywords for a document is statistically inferred, also has potential.