Literari.ly

a literary assistant

(working title)

Adam Lenart, Ankith Gunapal, Sandip Palit

Motivation

Many authors whether they write a book, a journal article or a blog struggle with a writer's block. We would like to help them by choosing the best expression that suits their target audience.

How do we help authors?

Authors would like to write manuscripts that people read in the present and in the future as well. A tempting way to assist them is to forecast the popularity of the expressions that they plan to use. Besides the popularity, it is also important to consider the context in which these expressions are being used and what sentiments are attached to them.

Data

Popularity

Google n-grams dataset it is a large, static dataset with the frequency of 1 to 5 word expressions in published books and magazines over time in multiple languages from about 1500 until now. Structured data, saved in CSVs, (list) partitioned by the starting letters of the expressions. On the other hand, some users might prefer unique expressions, and try to identify those that nobody else uses.

Sentiment/context-analysis

Google Books API can be used to find the volume which contains the queried expression. Based on this information, we can process reviews and ratings from Good Reads API.

Data availability

Google n-grams is a free dataset, Good Reads API and Google API are free upon registration.

Analysis

Popularity

Time series modelling. Google n-grams data contains the frequencies over time, by finding the most appropriate time series model, we can predict the future usage of that expression with appropriate confidence intervals.

Sentiment/context-analysis

Statistical analysis of the ratings that people give to the books. For more context, we can query the text of the reviews themselves: what do people like the book that this expression was published in? Do they use positive expressions? (note: we will need an external dictionary for classification).

We can also analyze the tags of the books, and look at the distribution of tags across topics and return it to the user, for example if the queried expression occurred 80% in technical publications, 15% in science fiction novels and 5% in romantic books.

Extra

- The expressions could be queried in the Twitter API and return the distribution of user interests who tweeted the queried expressions.
  Example implementation of finding out user interests at scale:
  http://www.mpi-sws.org/~mzafar/papers/recsys14-userinterests.pdf
  http://twitter-app.mpi-sws.org/who-likes-what/
- Fun features can also be added: how would a particular author use that expression by querying Google Books