Information Retrieval and Data Mining

19095174

February 2023

1 Task 1: Word Counter, Zipfs Law, Vocabulary list

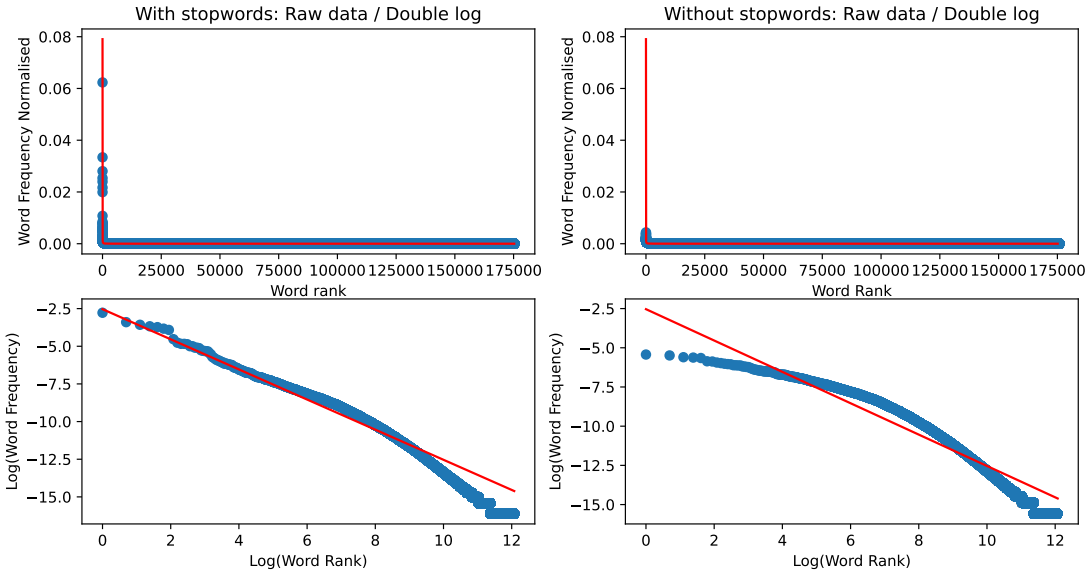
1.1 Preprocessing

Text Preprocessing steps used included lowercasing, removing non-alphanumeric characters (via string substituiton from re package) as during data exploration, there were many unusual punctuation that did not fall under the normal list of punctuation in commonly used packages, such as in *string.punctuation*, and a further removal of pure digits. A simple check after this first step still revealed many 'unhelpful' terms e.g. *'kmionosphere80', 'johnybhoi24', '219b', '10am5pm'*, in addition to many URL tokens that were deemed to not be helpful as they were too specific and infrequent, and therefore a second function was used to remove non-alpha terms. Lowercasing helps to combine the same words where some were captialised while others not.

The rationale was to keep the vocabulary size small, which may improve parsimony and efficiency for the tasks later on, when having to scan through the vocabulary. Upon brief inspection of the, I determined that many of the unusual words were not needed as they were too esoteric and to keep the vocabulary list to a minimal for more efficient lookup.

Qualitatively justify that these terms follow Zipf's law:

Figure 1: Plots of data (blue) against Zipf's Law (red)



The plots on the left show the normalised frequency against word ranking, without and with log transformation. It is clear in the double-log graph that the data closely follows Zipf's law, apart from the lower extremity, which quickly drops below Zipf's law, meaning that our lowest ranked words take up less probability mass than they are theoretically expected to. For the lower half, the gradient becomes more negative than -1.

Difference between the two distributions after removing stopwords

After removing stopwords, and renormalising the frequencies (to sum to 1), we can see that the data deviates significantly from Zipf's law - notably from the first plot on the right side that the highest ranked words fail to

capture the theoretical mass, hence losing the exponential drop in mass as rank increases. On the double log graph, the higher ranked words had datapoints that were below the straight line predicted by Zipf's law, while the middle ranked words were above the straight line, before dropping back down below the line again for the lowest words (similarly to without stopwords, showing that the low ranked words already struggled to capture the share they were theoretically expected to). This shows fit to Zipf's law is qualitatively worse after removing stopwords, presumably since it remove the highly ranked words that took up exponentially more share than the other words.

Before removing stopwords: the MSE between the log actual frequencies and the log Zipf's law frequencies was 2.51, while the MAPE (Mean Absolute Percentage Error) was 0.0972. After removing the stopwords, the MSE surprisingly dropped to 1.21, while the MAPE dropped to 0.0694, despite the fit looking qualitatively worse.

2 Task 2: Generating inverted index

I wanted to capture as much information in the inverted index that would assist in Tasks 3 (TFIDF, BM25) and 4 (Query Likelihood Models). These models required term frequency data, which could be captured in per-document granularity via 'term: (document id, term frequency)' key-value relationships. For example, in QLM, we need to know the number of times a (common) term appears in a specific document, and hence this extra information is crucial. Hence, I used a default dictionary (Collections) to store the (document id, term frequency) tuples for each word. Default dictionaries also do not require a word to be present first before initialising it in the dictionary, hence simplifies the code.

3 Task 4: Query Likelihood Models

3.1 Which language model expected to work better?

For an evaluation of model performance, since there are no given scores or 'target' values that indicate the true relevance scores of each passage, we will assess the models via subjective human evaluation.

I have created a function that randomly samples for a query, and then returns the top result returned for that query by all 3 models. The following is the first example:

Query 1: *What party is paul ryan in*

1. Laplace: *'when is matt matt ryan was born on the of may which was a matt ryan will be turning in only days from how old is matt matt ryan is years to be more the current age as of right now is minutes and how tall is matt matt ryan is m which is equivalent to what is matt zodiac matt zodiac sign is what is matt matt ryan does weight which is equivalent to where was matt ryan matt ryan was born in exton is matt ryan still matt ryan is still'*

2. Lidstone: *'thats right at least for the time being tea party darling sen rand paul is the effective leader of the republican party and thats a pretty big deal mo elleithee the gops shellacking in has thrust a new generation of republicans including leaders like rubio rep paul ryan sen ted cruz and govsbobby jindal and chris christie into the spotlight but its paul who has become the clearest voice of the new guard in the republican partyhats right at least for the time being tea party darling sen rand paul is the effective leader of the republican party and thats a pretty big deal mo elleithee the gops shellacking in has thrust a new generation of republicans including leaders like rubio rep paul ryan sen ted cruz and govs'*

3. Dirichlet: *'what political party does speaker of the house paul ryan belong to direct primary people vote to elect candidates from inside a particular political part used to select a single candidate from a political party closed primary only members of the political party can vote in the primary'*

The Laplace answer is completely irrelevant, picking out "Matt Ryan" instead of "Paul Ryan" as asked in the query. The Lidstone gives a relevant document "...new generation of republicans including leaders like ...rep paul ryan...", while Dirichlet returns a passage that contains "what political part does speaker of the house paul ryan belong to..." which in the context of query models, is effective, although the passage does not directly give the answer 'Republican Party'.

Query 2: *when is the month of autism awareness*

1. Laplace: *'each april autism speaks celebrates light it up blue along with the international autism community in commemoration of the united nationssanctioned world autism awareness day light it up blue is a unique global initiative that kicksoff autism awareness month and helps raise awareness about autismach april autism speaks celebrates light it up blue along with the international autism community in commemoration of the united*

nationssanctioned world autism awareness day light it up blue is a unique global initiative that kicksoff autism awareness month and helps raise awareness about autism'

2. Lidstone: **same result as Laplace**

3. Dirichlet: *'namely its the first day of national autism awareness month first given this designation in april by the autism society the goal of this month is to educate the public and build awareness around autism spectrum disorders and the difficulties and challenges that children with autism face'*

Laplace and Lidstone give the same answer, which answers the query less directly than the one returned by Dirichlet.

Query 3: *Question: what can contour plowing reduce*

1. Laplace: *'definition of contour plowing in the definitionsnet dictionary meaning of contour plowing what does contour plowing mean information and translations of contour plowing in the most comprehensive dictionary definitions resource on the web'*

2. Lidstone: *farmers can reduce soil erosion a increasing irrigation b contour plowing c grazing cattle on the land d plowing up roots*

3. Dirichlet: **same result as Lidstone**

Here, we see that Laplace gives the least helpful answer (definition of contour plowing but does not provide information), while Lidstone and Dirichlet are able to give a more meaningful answer ('reduce soil erosion').

Overall, we expect that Dirichlet-smoothed QLMs work better, as they allow for more information (e.g. words that are in the query but not in the documents can get higher probabilities in Dirichlet models) to be used.

Which language model expected to be similar?

The Laplace and Lidstone smoothing QLMs are expected to be similar, since they each increase the probability of all vocabulary words by a constant, hence only adjusts for information given by the relevant documents, whereas Dirichlet smoothing allows 'prior' information to be used by the entire corpus, including non-relevant documents, to give a 'general' probability of vocabulary words, hence increases the information and sample space used in generating the scores. As we have seen in the examples before, Laplace and Lidstone behave similarly relative to Dirichlet smoothed QLMs.

Comment on $\epsilon = 0.1$

This value of *epsilon* is not very small relative to the value of 1 used in Laplace (only one order of magnitude difference). This may not be a good choice, since the idea of adding *epsilon* is to give a very small but non-0 weight to relevant words that do not appear in the document.

Increasing μ from 50 to 5000

Considering that our average document length is only about 261 (after preprocessing, taken from BM25 calculation), increasing μ to 5000 may result in oversmoothing and a higher reliance on the probability of words given by the corpus, and less weight on the actual documents that we are trying to rank. For example, for an average sized document, a tiny weight of $260/(260 + 5000) = 0.05$ *approx* will be assigned to the MLE (Maximum Likelihood Estimate) of a word from the document data, while 0.95 weight assigned to the corpus MLE for the same word. Hence, this will be less appropriate than 50, which assigns more weight to the document MLE.