Information Retrieval and Data Mining (2024/25) Coursework 1

Anonymous ACL submission

Abstract

This document is a supplement to the general instructions for *ACL authors. It contains instructions for using the LaTeX style files for ACL conferences. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used both for papers submitted for review and for final versions of accepted papers.

1 Task 1

In this task, the vocabulary of the entire passage collection of raw text was obtained, where the vocabulary is the set of unique terms (uni-grams) extracted from the passage collection text after preprocessing. In addition, the frequency of each term (where a term is more formally known as a token) is also calculated after the raw text has been processed. These are both calculated in order to qualitatively justify whether the distribution of the frequency of terms follow Zipf's Law. The Zipfian Distribution is given by:

$$f(k; s, N) = \frac{1/k^s}{\sum_{i=1}^{N}(1/i^s)} \quad (1)$$

where k denotes the term's frequency rank in our corpus of text, s is the exponent characterizing the distribution, N is the number of terms in the vocabulary, s is a distribution parameter, and the denominator is the normalization factor.

Importantly, Zipf's law states that s is equal to 1 in the Zipfian Distribution. However, before obtaining the vocabulary and the term frequencies in order to analyse whether this corpus of text follows Zipf's Law, the pre-processing steps in order to obtain the necessary tokens must be stated. This was implemented using a custom text processor with the steps below explained, in order to split the text and format the tokens correctly.

1. **Reading the file:** The file 'passage-collections.txt' is read and the raw text inside this file is stored.

2. **Converting all raw text to lowercase:** This ensures that if any characters in the text are uppercase, they are converted to lowercase to standardise processing.

3. **Removing punctuation:** This removes any punctuation from the text according to the set: `!"#$%&'()*+,-./:;<=>?@[\]-‘{}|~`.

4. **Splitting text with consecutive alphanumerical characters:** Words with consecutive alphanumeric characters are split. An example of this is converting the raw text ["al27"] to ["al", "27"] or ["al27bn"] to ["al", "27", "bn"].

5. **Stemming:** This reduces words to their root form. An example of this is "running" to "run" or "jumps" to "jump". It must be noted that some irregular words such as "better" stay as "better" after stemming is applied. There are many different variations of the same root word. For example. "Running", "ran", and "runs" are all converted to the same root word "run". In this task, the Porter Stemming algorithm was used to achieve this. The implementation of this was accessed through the Python package gensim.

6. **Tokenise text by whitespace:** The tokens are obtained by splitting the text into individual words (1-grams) according to whether there is whitespace between the words.

7. **Removing stopwords from the tokens list (optional):** Stop-words such as "the" and "is" are removed from the set of tokens. The set of stop-words used are taken from the NTLK set of english stopwords.

Now, the text has been processed and tokenised using the steps above, the size of the index of terms, also known as the vocabulary, can be reported:

Stopwords Included	Stopwords not included
94365	94238

Table 1: Index size of the passage collection with Stopwords included and stopwords not included

The frequency of each term has been collected and the normalised frequency of each term against its rank can be displayed to determine whether this corpus of text adheres to Zipf's Law. Zipf's Law is presented as:

$$f(k) = \frac{C}{k} \tag{2}$$

Where $f(k)$ is the normalised frequency of the word at rank k, and C is a constant equal to $\left(\sum_{ni=1}^{N} \frac{1}{i}\right)^{-1}$.

In order to determine whether this corpus of text adheres to Zipf's Law, the logarithm of both sides of (2) will be taken and the gradient of the empirical data and the intercept will be calculated. This will be done via linear regression. The calculated value of C can then be extracted from the intercept by taking the exponential of the intercept. The calculated values of the gradient and C will then be compared to the theoretical values to determine whether this corpus of text adheres to Zipf's Law. First, when stopwords are included and second when stopwords are excluded from the analysed text. To illuminate this analysis further, I have included a derivation below

Zipf's Law is given by:

$$f(k) = \frac{C}{k} \tag{3}$$

Taking the natural logarithm on both sides:

$$\ln f(k) = \ln\left(\frac{C}{k}\right) \tag{4}$$

$$= \ln C - \ln k. \tag{5}$$

This shows that Zipf's Law follows a linear relationship in log-log space, where $\ln f(k)$ (log normalised frequency) decreases linearly with $\ln k$ (log rank), with a gradient of -1.

First, the calculated values of C and the gradient are considered when stopwords are included. The Empirical distribution of normalised frequency of terms against ranks are shown below. The corresponding plot of the natural logarithm of each of these is shown with the theoretical Zipf's law and the calculated line of best fit.

[display non-log plot]

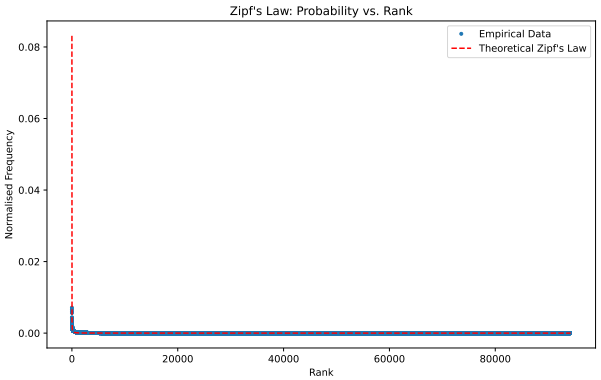

Figure 1: A plot displaying the normalised frequency of each term against the rank of each term. The Theoretical distribution of Zipf's law is included

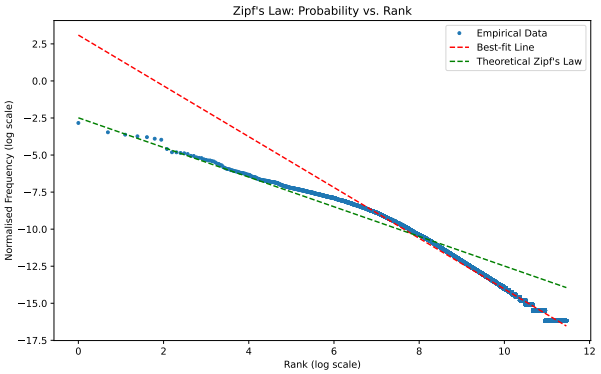

Figure 2: A plot displaying the natural logarithm of normalised frequency of each term against the natural logarithm of rank of each term when stop words are included in the analysis. A linear regression line of best fit is included as well as the theoretical distribution of Zipf's law.

...

The calculated values of C and the gradient are 22.1 ± 0.2 and -1.7 ± 0.1 respectively. The theoretical values are 0.08 and -1 respectively. The calculated values and their respective uncertainties suggest that the empirical data does

not follow Zipf's Law. However, analysing Fig. 2, it is evident that a simple linear regression might not be the best form of analysis to determine whether this corpus of text adheres to Zipf's Law. As seen, the line of best fit seems to be dominated by high rank terms which skews the gradient to be more steep and thus a vastly overestimated intercept value and underestimated gradient value. For log ranks lower than 8, the distribution of terms seems to qualitatively follow Zipf's Law.

Observing the results for when stopwords are excluded below, the calculated values of C and the gradient are 27.6 ± 0.2 and -1.7 ± 0.1 respectively. In addition, observing the distribution in Fig.3., it is clear that the higher rank terms dominate the linear regression fit, resulting in overestimated intercept values and underestimated gradient values again. Notably, there is a larger deviation from the theoretical Zipf's distribution from log ranks 6-8.

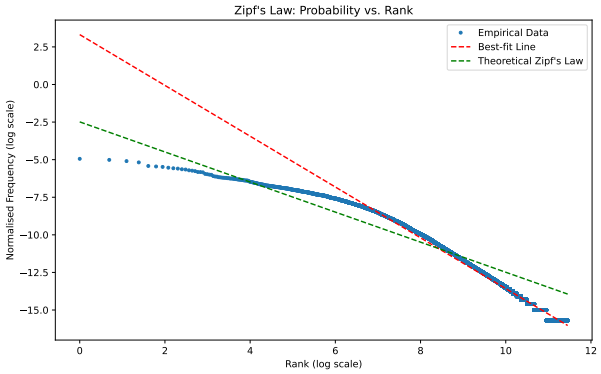

Figure 3: A plot displaying the natural logarithm of normalised frequency of each term against the natural logarithm of rank of each term when stop words are excluded in the analysis. A linear regression line of best fit is included as well as the theoretical distribution of Zipf's law.

Evidently, using using linear regression to judge whether these two distributions adhere to Zipf's Law, they both do not. However observing both Fig.2. and Fig.3., we can see that qualitatively they both adhere to Zipf's Law atleast in some portions of the two distributions.

When stop words are included, the empirical distribution of terms follows the theoretical distribution up until a slight hump near log rank 7. This is to be expected. Zipf's law contains several assumptions. Two of the most relevant for explaining the deviations in both Fig.2. and Fig.3. are the assumptions that the text Zipf's law is based on has access to an infinite vocabulary and is not exposed to any preprocessing steps (i.e. no stemming for example). This is why Fig.2. (when stop words are included) does not violate Zipf's law up until the hump; the highest normalised frequency terms are stop words which are taken into account by Zipf's law and these stop words are not stemmed by the stemming process. Therefore, their natural normalised frequency as predicted by Zipf's law is displayed. This is also the root of the reason Fig.3. displays a lower normalised frequency than expected for the initial ranks. Zipf's law assumes an access to an infinite vocabulary of which stop words are the highest occurring terms. However, in Fig.3. stop words are removed and the highest occurring non stop words naturally have a lower occurrence rate than stop words.

In both figures, a hump around log rank 7 is observed. This is suspected to be due to the stemming processing step. There may be a number of words which all have the same root word. This will artificially increase the term count of certain words which will not have the same term count in a natural, un-stemmed corpus of text. The ultimate effect of this is the 'hump' in the mid ranks which represent root words.

Lastly, the abrupt drop in normalised frequency in the tail of both figures, which skews the linear regression results, can be attributed to the invalid assumption of an infinite vocabulary. Zipf's law assumes an infinitely large corpus, but real-world datasets are finite. In an ideal Zipfian Distribution, there should be many instances of extremely rare words. But in this real corpus, there is not enough text and so the vocabulary is limited, which means not enough words are observed to match the power law desired. This is an effect which is present irrespective of whether stop words are included or not.

The distribution of normalised frequencies against ranks has been analysed, with and without stop words being included. The inconsistencies between the linear regression quantitative results and the qualitative observations have been attributed to a couple of

factors. Despite Fig.2. [check this] displaying a closer adherence to Zipf's Law, Stop Words will be removed for the following analysis. This is to improve computational efficiency and the assumption that stop words do not provide much meaning and will skew results for both retrieval and query likelihood language models.

[go into more reason on why this happens if have space.]

Stop-words will not included in the following

[need to add more...]

2 Task 2

An inverted index is a data structure used to map words (also defined as terms) to the passages in which they appear. For this task I extracted each unique instance of PID and its corresponding $passage$ from $candidate-passages-top1000.tsv$ and built an inverted index class which extracts a term in the vocabulary, the passages in which they appeared, and the frequency of the term in each passage. The exact data structure of the inverted index is:

- A **term** (word/token).
- A **posting list** that contains:
  - Document/Passage ID (`pid`).
  - Term Frequency (`tf`)—how many times the term appears in the passage.

For clarification, the data structure of the inverted index is a dictionary which is displayed below:

```
inverted_index = {
    "term_1": {pid_1: tf_1,
       ..., pid_n: tf_n},
    "term_2": {pid_3: tf_3,
       ..., pid_m: tf_m},
    ...,
    "term_k": {pid_x: tf_x
       ,..., pid_z: tf_z}
}
```

where:

- Each `term_i` represents a unique word in the corpus.
- Each `pid_j` is a passage or document identifier.
- Each `tf_j` is the frequency of the term in that passage.

In addition to this, I have calculated the length of each passage, where the length of each passage is defined as the number of tokens. This is stored in a separate dictionary with the structure:

```
passage_lengths =
    {pid_1: len_1, ..., pid_i:
        len_j}
```

where:

- Each `pid_i` is a passage or document identifier.
- Each `len_j` is the length of each passage.

These quantities have been stored for later tasks. For example, the inverted index is needed for TF-IDF, BM25, and Query Likelihood models because of its fast retrieval of term frequencies. The passage lengths are also needed for the tasks involving BM25 and query likelihood models.

3 Task 4

Query Likelihood models aim to rank passages based on how relevant they are to a query text. To be specific, documents are ranked based on the probability that they could generate the given query. The query likelihood model ranks documents by determining the probability of generating the query Q given a document D. This is formalised as:

$$P(Q \mid D) \tag{6}$$

Using the assumption that query terms are independent, this probability is computed as:

$$P(Q \mid D) = \prod_{w \in Q} P(w \mid D) \tag{7}$$

or in logarithmic terms for numerical stability and computational efficieny:

$$\log P(Q \mid D) = \sum_{w \in Q} \log P(w \mid D) \tag{8}$$

where $P(w \mid D)$ is the probability of term w appearing in document D. This is estimated using the term frequency in each document. The documents are then ranked by sorting them in order of decreasing $P(Q \mid D)$.

However, a pitfall of this general model is that, when a query term does not appear in a document, the entire probability becomes zero. This is unrealistic and smoothing offers a solution to this problem. The three smoothing methods explored are Dirichlet smoothing, the Lidstone correction, and Laplace smoothing.

Which language model do you expect to work better and why? Give a few empirical examples based on your data and experiments.

It is expected that Dirichlet Smoothing will work better than Laplace and Lidstone smoothing. As stated, all three of these smoothing techniques aim to account for the effect of an unseen term in a document. Dirichlet smoothing takes into corpus wide statistics while balancing the passage based probabilities. However, Laplace and Lidstone smoothing overestimate the probability of unseen words and do not consider corpus-wide term frequencies. These claims are made more apparent when the forms that these three smoothing techniques are presented below. First, Laplace Smoothing.

$$P(w \mid D) = \frac{\text{TF}(w, D) + 1}{|D| + V} \tag{9}$$

$$\log P(Q \mid D) = \sum_{w \in Q} \log \left(\frac{\text{TF}(w, D) + 1}{|D| + V} \right) \tag{10}$$

Where $\text{TF}(w, D)$ is the term frequency of term w in document D, $|D|$ is the total number of words in document D, and V is the vocabulary size of the general corpus.

Lidstone smoothing is presented below:

$$P(w \mid D) = \frac{\text{TF}(w, D) + \epsilon}{|D| + \epsilon V} \tag{11}$$

$$\log P(Q \mid D) = \sum_{w \in Q} \log \left(\frac{\text{TF}(w, D) + \epsilon}{|D| + \epsilon V} \right) \tag{12}$$

Where ϵ is a smoothing parameter typically in the range of 0 to 1.

Lastly, Dirichlet Smoothing is:

$$P(w \mid D) = \frac{\text{TF}(w, D) + \mu P(w \mid C)}{|D| + \mu} \tag{13}$$

$$P(w \mid C) = \frac{\sum_D \text{TF}(w, D)}{\sum_D |D|} \tag{14}$$

$$\log P(Q \mid D) = \sum_{w \in Q} \log \left(\frac{\text{TF}(w, D) + \mu P(w \mid C)}{|D| + \mu} \right) \tag{15}$$

Where $P(w \mid C)$ is a quantity known as the collection-wide probability of w and μ is a smoothing parameter which controls the weight of the corpus-wide smoothing.

Observing (9), it is evident that when a word is missing from a document (i.e. $\text{TF}(w, D)$ is equal to zero) Laplace smoothing assigns $P(w \mid D) = \frac{1}{|D|+V}$. This will inflate the probability of rare terms, ultimately making shorter passages artificially rank higher than they should. Though, Lidstone Smoothing slightly addresses this by incorporating a smoothing parameter ϵ, as seen in (11), this still does not completely address the issue of over-inflating the importance of rare words. Ultimately, this has the effect in both Laplace and Lidstone smoothing of over-rewarding shorter passages and penalising longer passages.

Dirichlet smoothing solves the problem of over-estimating the importance of rare words by accounting for their probability of occurrence in the general corpus, as seen in (14). The smoothing parameter, μ, controls how much one may want corpus wide statistics to have an effect on the smoothing. Observing (12), shorter passages get stronger smoothing while longer passages rely more heavily on term frequencies, which is a fair balance. This ultimately results in longer passages not being unfairly penalised when it comes to Dirichlet Smoothing, unlike Laplace and Lidstone smoothing.

[make sure to mention Stop words were excluded for computational ease]

3.1 References

The LaTeX and BibTeX style files provided roughly follow the American Psychological Association format. If your own bib file is named `custom.bib`, then placing the following before any appendices in your LaTeX file will generate the references section for you:

```
\bibliography{custom}
```

You can obtain the complete ACL Anthology as a BibTeX file from https://aclweb.org/anthology/anthology.bib.gz. To include both the Anthology and your own .bib file, use the following instead of the above.

```
\bibliography{anthology,custom}
```

Please see Section 4 for information on preparing BibTeX files.

3.2 Equations

An example equation is shown below:

$$A = \pi r^2 \tag{16}$$

Labels for equation numbers, sections, subsections, figures and tables are all defined with the `\label{label}` command and cross references to them are made with the `\ref{label}` command.

This an example cross-reference to Equation 16.

3.3 Appendices

Use `\appendix` before any appendix section to switch the section numbering over to letters. See Appendix A for an example.

4 BibTeX Files

Unicode cannot be used in BibTeX entries, and some ways of typing special characters can disrupt BibTeX's alphabetization. The recommended way of typing special characters is shown in Table **??**.

Please ensure that BibTeX records contain DOIs or URLs when possible, and for all the ACL materials that you reference. Use the `doi` field for DOIs and the `url` field for URLs. If a BibTeX entry has a URL or DOI field, the paper title in the references section will appear as a hyperlink to the paper, using the hyperref LaTeX package.

Limitations

Since December 2023, a "Limitations" section has been required for all papers submitted to ACL Rolling Review (ARR). This section should be placed at the end of the paper, before the references. The "Limitations" section (along with, optionally, a section for ethical considerations) may be up to one page and will not count toward the final page limit. Note that these files may be used by venues that do not rely on ARR so it is recommended to verify the requirement of a "Limitations" section and other criteria with the venue in question.

Acknowledgments

This document has been adapted by Steven Bethard, Ryan Cotterell and Rui Yan from the instructions for earlier ACL and NAACL proceedings, including those for ACL 2019 by Douwe Kiela and Ivan Vulić, NAACL 2019 by Stephanie Lukin and Alla Roskovskaya, ACL 2018 by Shay Cohen, Kevin Gimpel, and Wei Lu, NAACL 2018 by Margaret Mitchell and Stephanie Lukin, BibTeX suggestions for (NA)ACL 2017/2018 from Jason Eisner, ACL 2017 by Dan Gildea and Min-Yen Kan, NAACL 2017 by Margaret Mitchell, ACL 2012 by Maggie Li and Michael White, ACL 2010 by Jing-Shin Chang and Philipp Koehn, ACL 2008 by Johanna D. Moore, Simone Teufel, James Allan, and Sadaoki Furui, ACL 2005 by Hwee Tou Ng and Kemal Oflazer, ACL 2002 by Eugene Charniak and Dekang Lin, and earlier ACL and EACL formats written by several people, including John Chen, Henry S. Thompson and Donald Walker. Additional elements were taken from the formatting instructions of the *International Joint Conference on Artificial Intelligence* and the *Conference on Computer Vision and Pattern Recognition*.

References

Rie Kubota Ando and Tong Zhang. 2005. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6:1817–1853.

Galen Andrew and Jianfeng Gao. 2007. Scalable training of L1-regularized log-linear models. In

Proceedings of the 24th International Conference on Machine Learning, pages 33–40.

Mohammad Sadegh Rasooli and Joel R. Tetreault. 2015. Yara parser: A fast and accurate dependency parser. *Computing Research Repository*, arXiv:1503.06733. Version 2.

A Example Appendix

This is an appendix.