Coursework 1 Report

Runyang You
runyang.you.22@ucl.ac.uk

1 Task 1

1.1 Preprocessing

`findall` function of Regular Expression (re) is used to match the words in the given data. Symbols, numbers, and other non-English words are then deleted. Finally, `WordNetLemmatizer` from the `nltk` library are adopted for lemmatizing.

A total number of **108,359** different words were identified.

1.2 Comparison with Theory

The plots of probability of occurrence (normalized frequency) against frequency ranking are shown as Figure 1. Set $s = 1$ and $N = 116529$ in the Zipf's law

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

we obtain (approximately):

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

$$where \quad C = (\sum_{i=1}^{N} i^{-s})^{-1} \approx 0.08217$$

which is drawn as the 'theory' line in both plots.

From 1(b) it is obvious that before frequency ranking reaches 10^2, the two lines seen to be almost overlapping. But for data with frequency ranking between 10^2 and $10^3.5$ (approximately), the terms of same rank seen to occur more frequently, comparing with the theory. As for the ranking region of $[10^4, 10^5]$, as ranking increases and words become rarer, the words seen to appear much less frequently comparing with theory.

From 1(a) it is obvious that the two lines almost overlapped. Even though the data in 1(b) shows minor inconsistency with the Zipf's law as mentioned before, the two lines still follow similar trend, and thus we can arrive at the conclusion that these terms follow Zipf's law.

The difference can also be explained by the equation 1. the constant C is fixed given s and N.Then the normalized frequency of a term given by Zipf's law is strictly proportional to the value of C in log scale, which is not strictly obeyed by the given terms. For terms that rank in the middle range, the C values of which tend to be slightly higher than C. As rank increases, the C value of a certain word is gradually becoming smaller, even smaller than C. However, the overall trends of the 2 lines are relatively similar.

1.3 Removing Stop Words

Stop words are normally of high frequency in terms. Comparing a vocabulary with stop words and a one without, the value of the probability in the posterior vocabulary shall be significantly lower than the prior one, if given identical frequency ranking. Therefore, the difference between data and theory will increase, as shown in Figure 1(c).

The quantified difference are shown in Table 1, a total of 141 stop words are removed and resulted in a high MSE comparing with Zipf's law.

	# words	vocabulary	MSE
With stop words	10,242,703	108,359	**6.689e-09**
Without Stop words	6,019,331	108,212	**8.054e-08**

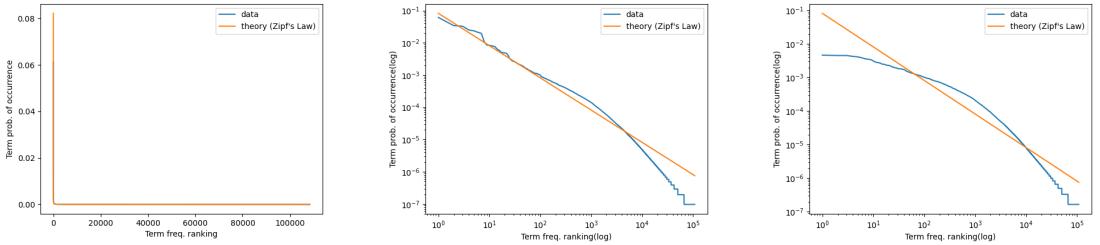
Table 1: Compared result of vocabularies

1.4 Data Storage for Further Processing

The vocabulary without stop words are then sorted by rank and stored with corresponding indices (rank number that starts from 0), as illustrted in Table 2.

0	1	2	3	4	5	6	...
one	name	wa	number	year	ha	also	...

Table 2: Stored Vocabulary



(a) Plot (with stop words) (b) Logarithmic Plot (with stop words) (c) Logarithmic Plot (no stop words)

Figure 1: Frequency against Frequency Ranking

2 Task 2

2.1 Inverted Index Generation

First, a datafrme (from `pandas`) is used for storing the mapping between `pid` and the natural indices starts from 0, along with the content of the passage, namely `passages_dataframe`, as shown in Table 3.

indice	pid	content
0	7130104	This is the definition of...
1	7130335	Best Answer: The AR ...
2	7130336	What does AR really m...
3	7130348	Oxycontin is generally...
4	8001869	STRATEGIC FEDERA...
5	8001870	Strategic federal credit...
6	7130674	Just reformat the cell m...
7	7130867	Binary (or base-2) a nu...
...	...	...

Table 3: illustration of `passages_dataframe`

A matrix M with shape (m,n) is then generated to store inverted index of all the given passages. $m = 108212$ is size of the vocabulary without stop words from Task 1; $n = 182469$ is the total number of all candidate passages. By using `dataframe.iterrows()`, each passage will be prepossessed with method from Task 1. Each value of the matrix $M[i, j]$, will then be set to the (unnormalised) frequency of the i^{th} word of the vocabulary, regarding the j^{th} passage, whose `pid` is `passages_dataframe[j].pid`.

Due to the sparsity of the matrix, the `lil_matrix` from `SciPy` is used when iterating passages and generating matrix M for efficiency. This matrix was later converted into `csr_matrix` for faster arithmetic multiply and product operations.

2.2 Data Stucture Explanation

Such way of generating inverted index directly stored the token frequency, while other crucial information such as passage length, normalized token frequency, document frequency can all be obtained by matrix operation methods provided by `SciPy`, e.g., `sum()`, `dot()` etc, speeding up the computation required by Task 3 and Task 4, since `SciPy` can do element-wise operations in parallel with its C-based framework.

3 Task 4

3.1 Better Model

Dirichlet should work better than the other two.

Take query "*university of dubuque enrollment*" (qid=532603) as an example, the top 2 retrieved passages by Laplace, Lidstone and Dirichlet are shown in Table 4. Lidstone and Laplace retrieved passages with frequently appeared term 'Dubuque', but the result is not really related to the query. Dirichlet however, retrieved 2 short passages, but contains 2 or more tokens in the given query, and of higher relevance.

3.2 Similar Models

Laplace and Lidstone smoothing are similar. Laplace can be seen as a special type of Lidstone smoothing with a fixed epsilon(ϵ) value equals to 1. Besides this parameter, the same information and computation process is adopted for obtaining the probability of a specific term. Examples can be found in Table 5. In contrast to Dirichlet, the Laplace and Lidstone models retrieved identical top three passages for the example queries.

	Top 2 Retrieved Passages of Query: "*university of dubuque enrollment*"	
Laplace and Lidstone	**Dubuque** County, Iowa. **Dubuque** County is a county located in the U.S. state of Iowa. As of the 2010 census, the population was 93,653. The county seat is **Dubuque**. The county is named for Julien **Dubuque**, the first European settler of Iowa. **Dubuque** County comprises the **Dubuque**, IA Metropolitan Statistical Area, and is the seventhlargest county by population in the state.	The **Dubuque** Police Department is the primary law enforcement agency for the city of **Dubuque**, Iowa. It is headquartered in the **Dubuque** Law Enforcement Center at 770 Iowa Street. This building is also home to the Emergency Communications Center, the **Dubuque** County Sheriff's Office, and the **Dubuque** County Jail. The **Dubuque** Police Department has 109 sworn officers and nine civilian employees.
Dirichlet	Complete the form below to request more information on admissions and financial aid from **University of Dubuque** * * * * * *	**Enrollment** of the 20 largest degree-granting college and **university** campuses: Fall 2014.

Table 4: Top 2 Retrieved Passages of Example Query: "*university of dubuque enrollment*"

	Top 3 Retrieved Passages								
Model $\backslash qid$	1108939			1112389			885490		
Laplace	8343919	5182924	3647358	5099991	5099993	7224811	664481	8323863	3468160
Lidstone	8343919	5182924	3647358	5099991	5099993	7224811	664481	8323863	3468160
Dirichlet	2555774	7121296	8630525	7117139	452837	452836	6942251	8667258	4434119

Table 5: Top 3 Retrieved Passages' IDs from Query 1108939, 1112389 and 886590

3.3 Comments on ϵ Value

Roughly speaking, the result of Lidstone smoothing can be seen as the probability of the modified document where each term appears additionally ϵ times. Therefore the parameter ϵ decides the extend to which the probabilities are smoothed.

The candidate passages have average length of approximately 33, while the queries have average length of 3.49, which suggests that a relative small ϵ valueapproximately $\frac{len(query)}{len(passage)}$ can be crucial for not over-shifting probabilities to the unseen terms (terms not appear in the query).

3.4 Comments on μ Value

For this problem we set $\lambda = \frac{N}{N+\mu}$, where N is the document length, we obtained:

$$\frac{N}{N+\mu}P(w|D) + \frac{\mu}{N+\mu}P(w|C)$$

Therefore, the value of μ decides how many weights (or how much trust) to give to the collection(the vocabulary),e.i., how many weights to take away from the document.

Notice that N is approximately 33, setting μ as large as 5000 means we are almost neglecting information provided by the document, and only trusting the collection, which cannot give more feasible results comparing with smaller μ value.