Information Retrieval Models for Passage Re-ranking

19138867
Department of Computer Science, University College London
London, UK

ABSTRACT

This project is to address the fundamental problem of short text retrieval and relevant passage ranking for text query in a information retrieval system. And the main goal of this project is to develop a text indexing model, inverted index, and two information retrieval model, vector space model and probabilistic information retrieval model. Statistical property of the terms count in a text dataset is measured and compared with Zipf's Law. The inverted index model is utilized in developing two basic information retrieval models, for reranking query's candidate passages dataset.

KEYWORDS

Information Retrieval, Text Indexing

1 INTRODUCTION

Information retrieval is an important infrastructure component among diverse kinds of applications, ranging from web search engine, library system to recommendation system. With the booming of data resource, information retrieval becomes much important than ever before that it is supposed to retrieve as much as possible documents/passages which are relevant to a given query. In order to increase the effectiveness and efficiency of retrieval process, text preprocessing and text indexing methods are also involved in this project. During the preprocessing, the non-word characters are eliminated and stemming is performed in each of query and passage for unifying the words which might have same meaning. Also, the vector representation is always used in representing a specific query/passages, but constructing a full vector representation is time inefficient and memory consuming. Inverted index can greatly increase the efficiency of text indexing and its input/output accessing as well as reduce the required memory space. Two traditional information retrieval models are introduced in this paper, vector space model, with using term frequency and inverse document frequency, and probabilistic information retrieval model, BM25. And the developed information retrieval model are tested under a query-passages candidate reranking dataset from the URL [1].

This paper will be formulated as the following sections. Section 2 is the literature review of different methods used for information retrieval model and the discussion of potentially pros and cons of the mentioned model. Section 3 will introduce the methodology used in the experiment, including text statistic, text preprocessing and indexing as well as the developed information retrieval models. In section 4, the experimental set up and the results obtained from the experimental methodology will be discussed. In the last section, it is the conclusion of this project.

[1] https://drive.google.com/file/d/1eKDfmDZoVuDADcR$_H$GMHMnjNJHDrXUs9/view

2 RELATED WORKS

There are various kinds of methods put forward to improve the relevance of the retrieved documents/passages in the information retrieval system including language model, clustering methods as well as the the recently developed attention based convolutional neural network.

2.1 Language Based Model

Language model has been used to improve the information retrieval performance. Oren et al. [2] building a hyperlinks, induced by the language model, among a chunk of retrieved documents and reranking the retrieved documents based on the calculated centrality of documents in the hyper-graph. Inspired by the language-based-model, the weighted directed hyperlink is assigned by the extent that the support received by a document d from a document o, $p_d(o)$. And $p_d(o)$ is measured by the the KL divergence of the maximum likelihood estimate of term frequency at document o in the retrieval collections and the term frequency of document d from the whole corpus collection using smoothed language model, e.g. Dirichlet-smoothed. Also, $p_d(o)$ can be measured by the cosine similarity between documents for simplicity. And PageRank or HITS (Hub and Authority) algorithms can be employed to calculate the centrality of the collection of retrieved documents. This methods can effectively provide a reranking documents/passages collection with combining the ideas from other domains smartly, such as graph analysis and natural language processing. But the efficiency of this algorithms would be degenerated when the weighted directed graph become larger and the dimension of the documents/passages representation is high. Although it can produce effective relevant results in a collection, this methods is not suitable to retrieve the relevant documents from a larger real world dataset.

Ensan[1] uses a similar idea, SELM, to construct a weighted undirected graph based on the semantics linking on the concepts between queries and documents. And the hyperlinks is defined and weighted between a query concepts and a document concepts, then the relation scores can be obtained from counting the semantic links in the query-document pair. Also, since it is not possible to define every words in the queries and documents as a concepts, this methods can be integrated with the well developed keyword-based retrieval system easily by learning the weights parameters through expectation maximization (EM) algorithm. And the final score for accessing the relation between a query and a document is the weighted sum of semantic links model and keyword-based model. However, the author mentions that SELM still has difficulty in getting optimal results when there is a concept which is not proper annotated.

2.2 Clustering Methods

Eilon [5, 6] recently proposes a nearest neighbour (NN) based methods to test the cluster hypothesis that "closely associated documents tend to be relevant to the same request" and developed a reranking methods for passages collection. The passages collection is firstly clustered by using NN methods, then the clusters is ranked by the learning-to-rank methods. In these paper, the relevance judgements in cluster between queries and documents/passages are relied on the relevance annotation from the dataset. And the predefined annotations are costly and scarce in the real world data. And, also, the NN methods might not keep its performance when the dimension of the feature vectors is high.

2.3 Attention Based Convolutional Neural Network

Attention based Deep Learning has draw people's attentions since its success in computer vision. And Yin et al. [7] introduced a convolutional neural network with attention mechanism, ABCNN, into Natural Language Processing, NLP, research on modelling the sentence pair. The attention mechanism is embedded into the input of CNN and the output of convolutional layer. ABCNN has outperformed the state-of-art deep learning model in various NLP research, especially in the answer selection task which is similar to the query's candidate passages reranking in this project. And there are information retrieval researcher continue work on this idea that Pathak et al. [4] utilized the attention mechanism in bi-directional LSTM, AQuPR, for passages retrieval. AQuPR improve the retrieval accuracy in most of given queries, but fail to retrieve passages for those queries where there are concepts in query which are not explicit annotated by the attention mechanism. As a result, AQuPR is not able to capture the syntactic structure, e.g. verb, adjectives and conjunction, of query and passage because the attention mechanism only focuses on matching the noun of query-passage pair.

3 TEXT INDEXING AND INFORMATION RETRIEVAL

In this section, the methodology used in this project are introduced, including the text preprocessing methods, text indexing and the information retrieval models, tf*idf vector space model and BM25 model.

3.1 Text Preprocessing

There are lots of non word character, such comma, underscore or semicolon, in the raw text data, including queries and passages, which might not represent any concrete meaning for the raw text data. Therefore, the non word characters in the raw text data are eliminated by regular expression in Python. The regular expression for eliminating the non word character is as following, which only keeps every character in query and passages belongs to any of lower case character, upper case character and number.

$$[\wedge \setminus sa - zA - Z0 - 9]+$$

And there are variants of a same words, which might or might not represent same or similar meaning, e.g. "like", "likes" and "likely". The stemming methods, Snowball Stemmer in this project, is used to address the variants of words, which is likely to change the original meaning of a word in the raw text data, depending on the stemmer methods used.

3.2 Zipf's Law

Many information retrieval models rely on the text statistic in the text dataset. And the occurrence counts of words/terms can be modelled as a distributions. The ranked term frequency follow long tail distribution, Zipf's law in natural language, where a few words occur frequently, and the others only occur a few times. And, in the Zipf's Law,

$$r * f = k \tag{1}$$

r is the rank of word's term frequency, f is the term frequency of a word and k is a constant parameter to be estimated. Equation (1) can be transformed to a linear model in log scale. The derivation is as following

$$\begin{aligned} r * f &= k \\ log(f) &= -log(r) + log(k) \end{aligned} \tag{2}$$

Therefore, the constant parameter, k, can be obtained from a maximum likelihood estimate with using the empirical data with length |N| by linear regression, where

$$\begin{aligned} \sum_i^N log(f_i) &= \sum_i^N (-log(r_i) + log(k)) \\ \sum_i^N log(f_i) &= N * log(k) + \sum_i^N -log(r_i) \\ log(k) &= \frac{1}{N} \sum_i^N log(f_i) + log(r_i) \end{aligned} \tag{3}$$

In statistic, the coefficient of determination, R^2, can used to access if the empirical data can be explained by the linear regression model with the estimated constant. And the R^2 is calculated by

$$\begin{aligned} TSS &= \sum_i^N (y_i - \overline{y}) \\ RSS &= \sum_i^N (y_i - f_i) \\ R^2 &= 1 - \frac{RSS}{TSS} \end{aligned} \tag{4}$$

y_i is the value of data i in the empirical data, $\overline{y}$ is mean value of empirical data and f_i is the predicted value of data i from the estimated linear model in equation (2). The R^2 is ranging from 0 to 1. And if the R^2 is close, the linear regression model we get can better explain the empirical data from the experiment.

3.3 Inverted Index and Tf-Idf Representation

Constructing the full vector representation, which gather all the words occurring in the raw text data as the vector entries and count the occurrence of each terms in each of passage as the values of each entry in the vector representation, is inefficiency in program

running time and memory consuming. Inverted index is used instead, and it stores the count of occurrences of each words with indexed by each passage in an inverted list. Therefore, inverted index does not need to check if a specific word appear in a specific passage for every entry of the corpus of dataset. Instead it only need to distribute the terms count of each passage to the corresponding entry in the inverted list. And the dictionary data structure is an ideal container for inverted index model which greatly increase the search time of text statistic.

Term frequency, TF, is the number of time that a word/term occurs in a passage. And, in my implementation of the inverted index, every entry stores TF of a word/term, indexed by the an passage, the total occurrences of the word in the collection and the inverse document frequency of the word for the collection. The inverse document frequency, IDF, demonstrate a word in discriminating passage in the dataset. The word occurs in more passage of the dataset, the less ability the word have to discriminate documents. Therefore, IDF is used to increase/penalize the weights, TF in this paper, used in information retrieval. IDF is calculated by

$$IDF_t = log_{10}(\frac{N}{n_t}) \tag{5}$$

N is the number of passages in the collection and n_t is the number of documents in which the term occurs.

3.4 Vector Space Retrieval Model

Information retrieval of passages with given query is made by measuring the similarity between a query and the candidates passages. The queries and passages are represented as vector, where the query vector is the binary full text representation of collection and the passages vector stores the TF*IDF text representation of the collection. And the similarity score of a query, q, and a candidate passage, p, with cosine normalization is calculated by

$$sim(q,p) = \sum_{t \in (q \cap p)} \frac{TF_t x IDF_T}{\sqrt{\sum_{i \in q} TF_i x IDF_i}} \tag{6}$$

3.5 Probabilistic Retrieval Model, BM25

The BM25 model is developed from binary independence model, $\frac{P(D|R)}{P(D|NR)}$. $P(D|R)$. $P(D|R)$ is the probability distribution of relevant passages, and, similarly, $P(D|NR)$ is the probability distribution of non-relevant passages. The probability distribution is assumed to be independent $P(D|R) = \Pi_{i \in D} P(d_i|R) = \Pi_{i \in D} P_i$. Based on the binary independence model, the score of a passage to be relevant is derived as in Christopher D et al. [3]

$$sim(q,p) = \sum_{t \in (q \cap p)} log(\frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)}) * \frac{(k_1+1)f_i}{K + f_i} * \frac{(k_2+1)qf_i}{k_2 + qf_i} \tag{7}$$

r_i is the number of retrieved relevant passages, R is the total number of relevant passages, n_i is the total number of retrieved passages, N is the number of passages in the collection, f_i is the term frequency of term i in the collection and qf_i is the term frequency of term i in the given query. And K is set empirically with k_1 set as 1.2, k_2 set between 0 and 1000, 100 in this project, and b set as 0.75.

$$K = k_1((1-b) + b * \frac{dl}{avdl}) \tag{8}$$

dl is the length of the passage which is being measured the similarity with a given query and avdl is the average passage length in the collection.

4 EXPERIMENT RESULTS

4.1 Text Statistic and Zipf's Law

The text statistic of corpus in the dataset *passage_collection_new.txt* from [2]. The non-word character will be eliminated from the raw text data, including queries and passages, then the terms/words is stemmed by using Natural Language Processing Toolkit, Snowball Stemmer. The language stemming method is possible to change the meaning of the original query and passage, such as both "like" and "likely" will be stemmed as "like". The text statistic, probability of a term occurring, is compared with the Zipf's Law , with parameter, α of Zipf's distribution being 1, is in Figure 1. The reason why choosing α being 1 is that it is a degree of 1 polynomial in equation(2), and the Zipf's Law assumes that the probability of rank t term is $\frac{1}{t}$ of the first rank term.

Since the term occurrences probability follow a long tail distribution, and there are lots of words which only occur a few times. In order to enhance the visibility, there are only the top 150 occurrence terms plotted in the Figure below. The maximum likelihood esti-

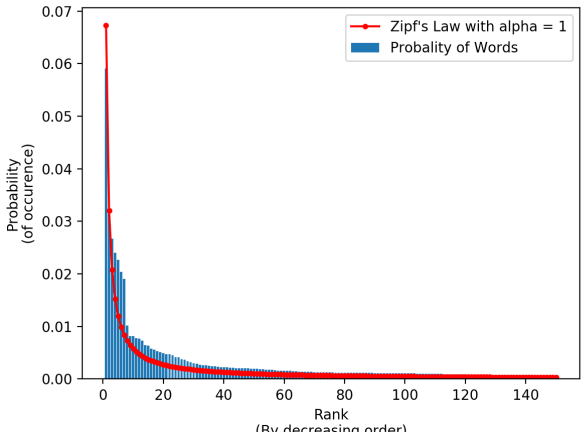

Figure 1: Text Statistic

mation of the bias term in equation (3) is 12.21. And the calculated coefficient of determination is 0.82, which indicates that the Zipf's Law is able to describe the occurrences probability of terms. The result is shown in Figure 2.

[2] https://drive.google.com/file/d/1eKDfmDZoVuDADcR$_H$GMHMnjNJHDrXUs9/view

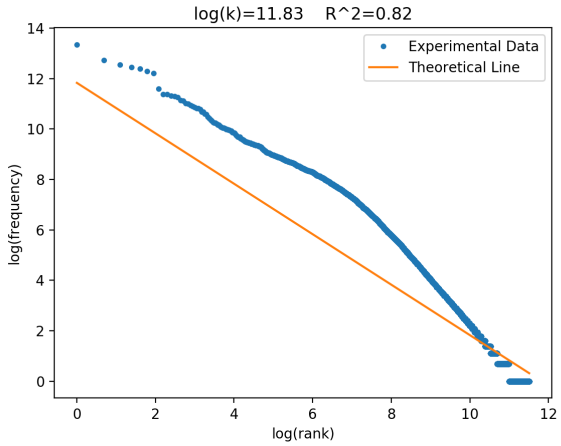

Figure 2: logarithm plot of equation (3)

4.2 Text Indexing in Retrieval Model

The text statistic is stored by using inverted index for each of query's candidate passages collection. And the inverted index stores the term count of each passage, the inverse document frequency, IDF, of a term in the collection and the total counts of a term in a collection for quickly accessing of statistical data to calculate the similarity score between query and collection. In this project, the candidates passages of every query are assigned an inverted index separately.

And the inverted index can get around the time inefficiency and memory consuming problem that full vector space representation has. Also, the text statistics is stored in Json files, which has a data storage format of key-value pair, such that it is more compatible to the inverted index data structure. When it comes to calculating similarity scores for executing the information retrieval, the relevant statistics of a term in the query's candidate passage collection is invoked by using its corresponding query ID and passage ID. Since the similarity scores is formed by the score of common terms between query and passage, the corresponding similarity scores can be obtained incrementally, which is equivalent to the dot product of vector representation in vector space model.

4.3 Reranking Passages

The query with its candidate passages collection is from dataset $candidate\_passages\_top1000.tsv$ under same directory of the dataset used in the Text Statistic section. Most of the query have more than 100, typically 1000, candidate passages. And the top 100 reranking are used as the final output of the retrieval model. There a few queries that have candidate passages less than 100, such that all the reranking results are used as final output. The reranking results of vector space model is $VS.txt$ under the submission folder. And the results for BM25 is $B25.txt$ under the same folder as the results of vector space model.

For BM25 model, Since there is not any relevance information which indicates the relation between query and its candidate passages. Therefore, the parameter r_i and R in (7) are set as 0 for BM25 model. The others parameters in (7) can be accessed through the developed inverted index by using corresponding query ID and passage ID. Also, in order to access the passage length, dl, and average passage lenght, avdl, in a collection for (7), the statistic of that two parameters is also collected during the text preprocessing stage and stored in another Json file indexed by each of query ID.

5 CONCLUSION

This project is to analyze the statistical property of the terms in passages collection, implement a efficient text indexing model, inverted index, and developed two information retrieval model, vector space model and BM25 model, for candidate passages reranking for a query. From the experiments, the empirical statistics of terms in the passage collection can be explained by the Zipf's Law based on the coefficient of determination. The inverted index can greatly increase the efficiency of access the text statistics compared with developing a full vector space representation for passages. And the candidate passages for query are reranked by the similarity scores given by the vector space model and probabilistic model, BM25. The top 100 results are reported as the final output of this passages reranking project.

ACKNOWLEDGMENTS

This report is for the Assignment 1 of COMP0084 Information Retrieval and Data Mining.

REFERENCES

[1] Faezeh Ensan and Ebrahim Bagheri. 2017. Document Retrieval Model Through Semantic Linking. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining* (Cambridge, United Kingdom) *(WSDM '17)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/3018661.3018692

[2] Oren Kurland and Lillian Lee. 2005. PageRank without Hyperlinks: Structural Re-Ranking Using Links Induced by Language Models. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (Salvador, Brazil) *(SIGIR '05)*. Association for Computing Machinery, New York, NY, USA, 306–313. https://doi.org/10.1145/1076034.1076087

[3] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval.* Cambridge University Press, USA.

[4] Parth Pathak, Mithun Das Gupta, Niranjan Nayak, and Harsh Kohli. 2018. AQuPR: Attention Based Query Passage Retrieval. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 1495–1498. https://doi.org/10.1145/3269206.3269323

[5] Eilon Sheetrit and Oren Kurland. 2019. Cluster-Based Focused Retrieval. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (Beijing, China) *(CIKM '19)*. Association for Computing Machinery, New York, NY, USA, 2305–2308. https://doi.org/10.1145/3357384.3358087

[6] Eilon Sheetrit, Anna Shtok, Oren Kurland, and Igal Shprincis. 2018. Testing the Cluster Hypothesis with Focused and Graded Relevance Judgments. In *The 41st International ACM SIGIR Conference on Research Development in Information Retrieval* (Ann Arbor, MI, USA) *(SIGIR '18)*. Association for Computing Machinery, New York, NY, USA, 1173–1176. https://doi.org/10.1145/3209978.3210120

[7] Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2015. ABCNN: Attention-Based Convolutional Neural Network for Modeling Sentence Pairs. arXiv:cs.CL/1512.05193