COMP 4651 Cloud Computing and Big Data Systems - Project Report

Analysis of Topic Evolution in Google Books

Group Members: CAI, Fengyu (20256291), CHEN, Ziqi (20176623), SHI, Yifei (20255780), ZHANG, Zizheng (20256796)
Supervisor: WANG, Wei

Abstract

With the rapid development of information technology, there are more and more raw data stored on the Internet. These information is valuable because it shows the trend of the social development and the habits of human beings in many different areas. What we need to do is to apply the technical of *big data* and *cloud computing* and dig up the deeply hidden information.

This project aims to find the topic evolution pattern by analysing the data from the *Google Books N-gram Dataset*. We apply the frequency analysis and correlation analysis to some of the hot topics nowadays. With the help of the AWM EC2, we could easily use the Python and Spark to obtain and analyse a large amount of data. The results show that many words do have a fixed evolution pattern and there are many hidden interconnections between some words which are not obvious. The results provide sound evidence for the social analysis research.

Introduction

Understanding how the topic evolves in the books is an interesting and important topic in the field of natural language processing. The boom of big data provides a new approach for the linguistics study, and the progressively increasing computing capacity also makes it possible to analyse massive stored material. With these new research methods, we try to figure out the linguistics evolution of specific topics in the past few decades (Single term analysis) by digging up the dataset, like Google Books, and further, we would like to reveal the potential relationships between certain representative terminologies (Covariance analysis).

The report will describe the project in detail and can be divides into sections as follow. Section II is to briefly describe the work done by different group members. Section III will show the methodology of the project, including the software we used, the project process and the

sample codes for data analysis. Section IV will show the analytical results together with printed graph, which makes it easier to find the evolution pattern of certain Bi-grams. Section V will explain the challenge exists in this project and potential development of the future.

Division of Work

There are four members in this project. CAI, Fengyu and ZHANG, Zizheng are in charge of the data capture. SHI, Yifei works to analyse the frequency of occurrences of the Bi-gram in the books. CHEN, Ziqi's job is to analyse the correlations between the Bi-grams and show them with the graph.

Implementation Details

Pre-processing

This part of the project aims to find the raw data for analysing. The resource we used is mainly from the Google Books (Link is attached at the end of this report). The data pattern is as follow:

```
ngram TAB year TAB match_count TAB volume_count NEWLINE
```

The data is sorted by the initial of the words and stored in different files. The average size of the data file is about 2GB and there are many data we do not need, so we have to use the cloud computing method to dig up the data we are interested in. We write a Python program and run it on the EC2 which finally give out the data we wanted and store them into a text file for future analysis.

The following is the sample code of data digging up.

Code:

```
def get_list():
        with io.open('wordlist.txt', 'r') as file:
                wordlist = file.readlines()

        wordlist = [i.replace('\n', '') for i in wordlist]
        #print(len(wordlist))
        final_list = []
        for name in wordlist:
                final_list += preprocess(name)

        output = open('output.txt', 'w+')
        for i in final_list:
                output.write(str(i))

        output.close()
        #print(len(final_list))
        return final_list
```

Frequency Analysis

Methodology

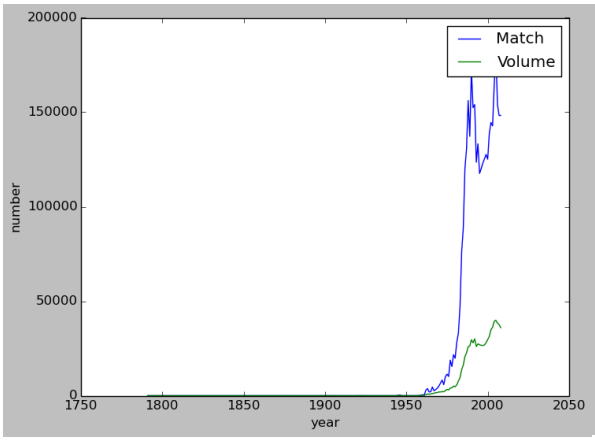
For this part, we would like to find the frequency of occurrences of the bi-grams. Considering the complexity of the analysing, we apply the Spark in this part because there are some useful functions for data analysis in the Spark. We would like to analyse the data in two ways. One is to find the frequency of occurrences of different years since 19th century (Some of data are even earlier, which can be ignored in big data analysis). The other one is to find the frequency of occurrences in different books. It counts that how many books in total that include the target bi-grams.

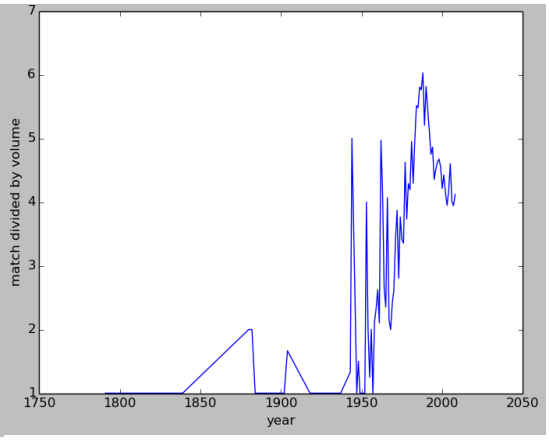
A sample block of code is attached below.

```
def match_divide_by_volume(bigram):
    new_bigram = bigram.map(lambda x:((x[0],x[2]),(x[3],x[4])))
    reduced_bigram = new_bigram.reduceByKey(lambda a,b:map(sum,zip(a,b)))
    mdbv = reduced_bigram.map(lambda x:(x[0],float(x[1][0])/x[1][1]))
    return mdbv

mdbv = match_divide_by_volume(bigram)

top_15=mdbv.takeOrdered(15,lambda x:-x[1])
print top_15
```

Results

We explored the change in frequencies of a few words, as a demonstration, below we plotted the volume and page occurrences for the bi-gram "artificial intelligence". We can see that the bi-gram showed a burst in frequencies starting from around 1990, dropped down for some years, and then rose again after 2005. This matches with our knowledge – there are rapid changes and improvements in the research of artificial intelligence in the periods mentioned above.



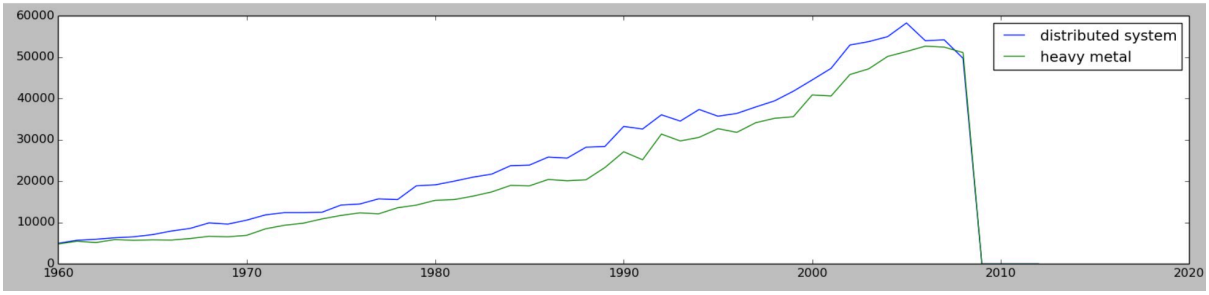


Correlation Analysis

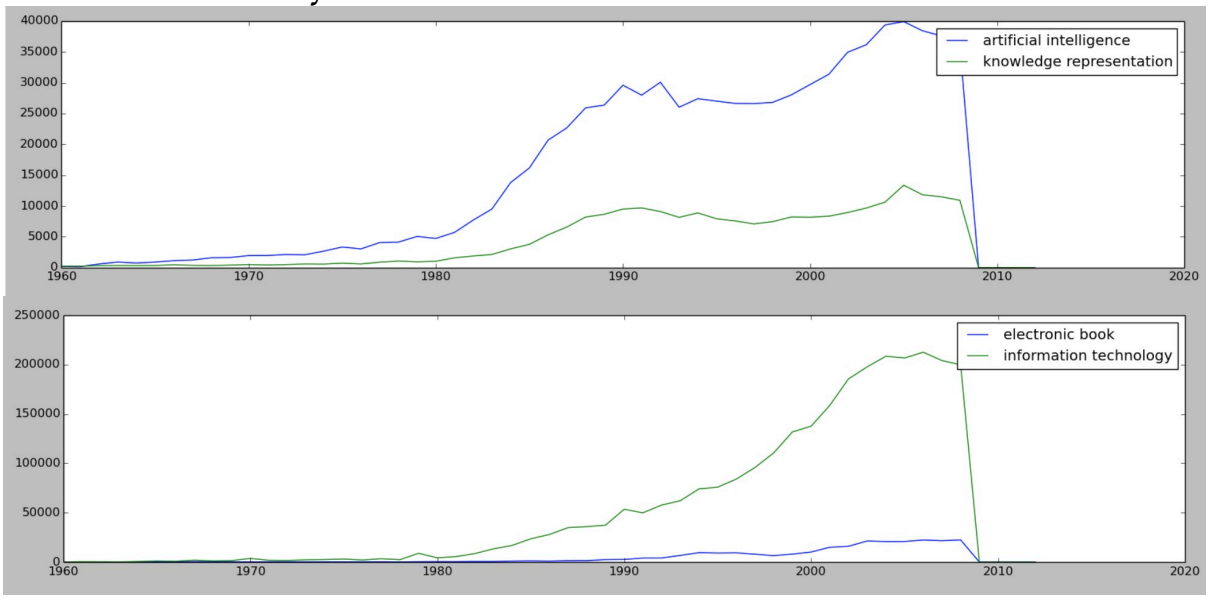
For this part, we would like to explore the interconnections between different bi-grams by computing the similarity in occurrences in different years. To do so, we apply the following steps. First, we compute the number of volumes and pages where a bi-gram occurred in each year, and obtained a RDD with the bi-gram word as key, and the {year: count} dictionary as values. Then we do Cartesian product on the RDD, and compute the dictionary similarity or dissimilarity between pairs of bi-grams. There are three metrics for similarity/ dissimilarity are:
(1) L1-distance, i.e., mean absolute difference.
(2) L2-distance, i.e., mean squared difference.
(3) Cosine similarity, dot_product (dict_A, dict_B) / (norm(dict_A) * norm(dict_B)).
After computing the similarity/dissimilarity between pairs, we sorted the results and picked several top results.

Results

We found that results given by L1- and L2-distances were sometimes hard to interpret. For example, the bi-gram "barack obama" demonstrated a similar trend to "open source" in volume occurrences. The same happens between "distributed systems" and "heavy metal". There might be correlations in these bi-grams, however, the relations between them may be hard to reason about.



We also found that results given by cosine similarity showed more reasonable interconnections between bi-grams. For example, we found that the pair of "artificial intelligence" and "knowledge representation" showed high similarity, as well as the pair of "electronic book" and "information theory". These results are more intuitive.



Conclusion

The results show that there are many hidden interconnections between some words which are not obvious. These sound evidence can be used for the social analysis research. For example, the "integrated circuits" and "natural language" have very obvious connection but the reason why it works in this way still need to be researched in the future.

Useful Link

Link of Bi-gram data:
http://storage.googleapis.com/books/ngrams/books/datasetsv2.html
Github Link of Project:
https://github.com/kaychenziqi/google-book-ngram-analysis