CSCI 8876 Final Project
Kwok Sun Cheng
04/27/2020

Topic modeling with bioinformatic literatures

- **Abstract**

  As the world wide web grows rapidly, text corpus is becoming increased online at an incredible rate. Managing a corpus of documents is critical for many areas of science, industry, and culture. For example, bioengineering researchers, who study a new generation of advanced materials, frequently need to identify and understand a comprehensive body of literature describing an association between material features of interest. However, there is no inspection technique to help such researchers who need to make critical decisions based on their understanding of a corpus of documents. In my project, I will present an application I have developed. Users not only use it for literatures searching. They also are able to use the topic modeling technique to help them extract some topics in order to find out more relevant field with the keywords they are interested in.

- **Background**

  Vast amount of document collections is becoming available in large repositories with the rapid growth of hardware platform and soft- ware technology for the world wide web. The National Center for Biotechnology Information (NCBI) repository (https://www.ncbi.nlm.nih.gov) provides millions of bibliographic documents [1]. Many researchers often need to inspect these large document collections to understand datasets and make a critical decision. For example, when some bioengineering researchers explore underlying biological mechanisms of biofilms, their activities need to frequently identifies and understands a comprehensive body of published literature studying and identifying relations between material features of interest.

  However, these researchers typically have a great difficulty exploring such ever-growing big datasets when they survey and evaluate new information in the existing bioengineering literature. For example, their questions, “Which are the useful information?” or “Which are repeatedly used in the context of different kinds of documents?” on a text corpus cannot be often answered.

  The increasing amount of text data creates a need for advanced approaches that can learns interesting and important patterns from the data. Structured data can be managed by a database; however, for unstructured text data, approximate keyword searching, and random browsing are usually used to manage and find useful information from a collection.

  In my database, it contains some bioinformatic literatures I have collected from PubMed and PMC. The language I have used is MySQL which means my database is a relational database. It has three entities and two relationships. Figure 1 is my Entity Relationship Diagram (ERD).

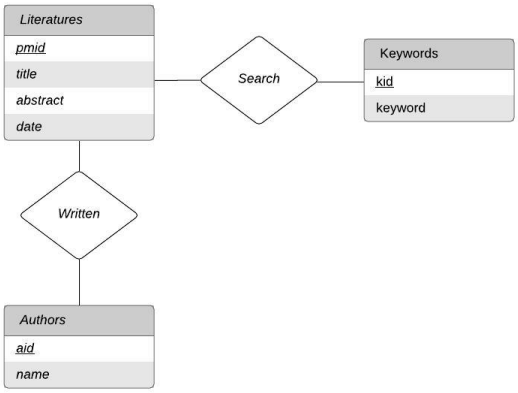

Figure 1: ERD

- **Research Question or Goal:** The Goal of building this database is researchers can use my database to archive their research quickly and conveniently. Researchers can easily just type a keyword to search what literatures they are looking for. They also can use it to find some potential topics which relate to the keyword they searched by using topic modeling technique.

- **Objectives:**
  - **Literatures Searched**
    First thing researchers will have from my database is a list of literatures relate to the search keyword. They can see literatures' id and title from the list. If they see an interested title, they can just type the paper's id, and it will show them more detail about the paper such as abstract and writers' name.
  - **Topic modeling**
    Topic modeling is a powerful technique for researchers to analysis a huge collection of data. Researchers can use it to discover some hidden title for the data [5]. After literatures searching, researchers can also use the result to do the topic modeling. They can go back to the main menu and select topic modeling. Then, typing in a keyword they are interested in and a number of topics. After that, the program will search the literatures from the database to do the topic modeling. Once the program gets done, it will show the result with a list of topics with some terms. Researchers might find out some interesting topic by observing the result. If they type the topic number, it will show them more terms.

- **Methods:**
  - **Data Description:**
    - **Data obtained and its source**
      My data's source is from PubMed and PMC. I have been using NCBI API, E-utilities, to get my data from PubMed. The data I've been collecting are id, title, authors, published date and keywords.
    - **Specs of the data obtained**
      I will store the data as json format first, then write a script to insert the data into database. Each json file contain maximum 400 papers' data because of json size limitation. Figure 2 is a format of json file. I've already downloaded around 30000

papers by using a keyword, biofilm. I've already written a script to insert those data into the database.

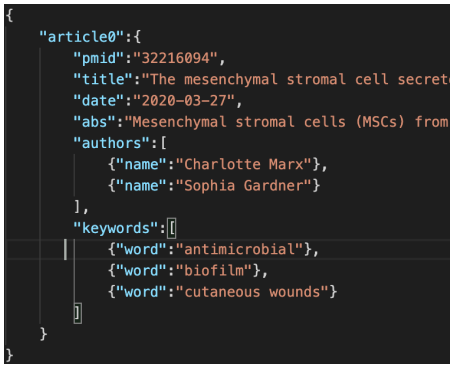

Figure 2: JSON Format

- **Methodology**
  - **Creating DB**

    First, downloading data from PubMed and PMC by searching some keywords using E-utilities. In this step, I'm using Python to do that. I wrote a program to get two input, keyword and number of results. Then, using those input to create an URL which is E-utilities to download the xml format result. After downloading, searching information I need, reformatting them as json format, and then storing the result into a data folder. It's my methodology of data collection. In this step, all source codes are at the directory, workspace/database.

    After collecting data from PubMed and PMC, I wrote a program to insert all data I have collected into my database using MySQL and Python. First, my program read all JSON files and decomposed them to different data. Then, it inserted data into entities, literatures, keywords, and authors. After that, searched the keyword's id, kid, and authors, aid, in order to create relationship of literatures-keywords, search, and literatures-authors, written. By the way, literatures' ids are the id of PubMed or PMC, and kid and aid are auto id.
  - **Running Application**

    In the application, once user run the program, it will show an interface which will ask for three option input, 1. Literatures searching, 2. Topic Modeling, and X. Exit the application.
    - **Literatures searching**

      When user select first option, the program will ask a keyword which they are interested in. After typing in the keyword, the program will use the keyword search the database for relevant papers and display the result as a list which contain paper ids, titles, and published dates. If user see a title they are interested in, they can type the paper id, and the program will show them a paper view which contains more detail about the paper such as abstract, authors, and keywords list.
    - **Topic modeling**

      When user select second option, the program will ask a keyword as same as first option. However, it will also ask user for a number of topics. The program will use this number and keyword provide a result of topic modeling. In this process, the program might take some time because it will need to go through few steps before doing the topic modeling. First, the program will collect

dataset from the database by using the input keyword as same as searching literatures.

Second step is preprocessing the dataset, and it also is my project start going outside the scope of my knowledge from this step. Many topic modeling research studies the impact of the preprocessing task which is one of the key components and often has an impact on the experiment results [3, 4]. the preprocessing stage consists of several steps such as tokenization, filtering, and lemmatization. The import part in this step is lemmatization. The program applies lemmatization to mitigate the weight of the terms in the document collection and to match distinctive terms (words or phrases) effectively. For example, {am, are, is} is normalized to {be}. To make texts comparable, unnormalized text is segmented into tokens by keeping together alphabetic characters and all other (non-space) characters are separated into their own tokens.

Then, creating dictionary and bag of word (BoW). The program will use the preprocessing data to create the dictionary first. After that, modeling each of the documents by computing the number of times each term is seen. The bag of words model representation is used in information retrieval and natural language processing. In BoW, it represents text by ignoring its order and grammar [5].

Finally, applying the number of topics, BoW and dictionary to train the topic model. In this step, the program is using one method of topic modeling which is Latent Dirichlet Allocation (LDA). LDA is an algorithm which can process huge text data, and it is a Bayesian Hierarchy model, in which a set of text data is modeled as a mixed model of various topics [6]. After getting the model done, the program will display the result with a list which contains topic number and top 5 terms which has highest weight in the topic. If user type the topic number, the program will show a view which contain more terms.

- **Code**

  I save all my source codes into GitHub repository. In the repository, it has three directories, workspace, application and database. In database, I store my DDL code. In workspace, I stored my Python codes and json format data.

  In the workspace folder, I also separate into two parts, database and web. Database folder contain all the source codes for collecting and creating my database. Application folder contain the main program which is my core of my project, search and topic modeling engine. Web folder contain the source codes for the web application, but it's no completed yet.

  - **DML and DDL:**
    https://github.com/kcheng18/CSCI8876/tree/master/database

  - **Workspace:**
    https://github.com/kcheng18/CSCI8876/tree/master/workspace

- **Results**

In the literatures searching, the program successfully gets the user input, in this case is biofilm, and display list of literatures to them with id, title, and published date, figure 3. From figure 5, it also can display more detail about the literature.

In the topic modeling, the program has also done well with display the result of topic modeling, figure 4. Form figure 6, we can know what is about the topic 1 by observing some terms from the view such as plant, water, lifestyle, etc. Therefore, researchers can use this information for the research.

```
Kwoksuns-MacBook-Pro:application kwoksuncheng$ python3 main.py
[nltk_data] Downloading package stopwords to
[nltk_data]     /Users/kwoksuncheng/nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[1]    Papers search
[2]    Topic Modeling
[X]    Exit
Please enter your option:
1
-------------------- Papers search --------------------
Please enter the keyword you want to search:
biofilm

5698 Found
|    Pmid       |                 title           |   Pub_date    |
|   23831146    | Which metaproteome? The im... |  2013-09-09   |
|   23831189    | Effects of local delivery ... |  2013-07-22   |
|   23831483    | Zingerone inhibit biofilm ... |  2013-10-07   |
|   23831592    | Nafion coated stainless st... |  2014-09-15   |
|   23831746    | Performance of aerated sub... |  2013-07-26   |
|   23831793    | Assessment of toxicity thr... |  2013-09-09   |
|   23834327    | The effects of mechanical ... |  2018-12-02   |
|   23836904    | MmpL11 protein transports ... |  2018-11-13   |
|   23837889    | Oral microbial colonizatio... |  2013-08-07   |
|   23840194    | The Human Cathelicidin Ant... |  2018-11-13   |
10/5698
```

Figure 3: Literatures Search

```
[1]    Papers search
[2]    Topic Modeling
[X]    Exit
Please enter your option:
2
-------------------- Topic Modeling --------------------
Please enter the keyword you want to search:
biofilm
Enter the number of topic (1-20):
10
Running Time for setting up dataset:  2.0960779190063477 seconds
Running Time for building model:  0.9835419654846191 seconds
Data preprossing.............
Running Time for preprossing dataset:  18.362335920333862 seconds
Running Time for generating corpus:  0.3096017837524414 seconds
Running Time for topic modeling:  14.126457929611206 seconds
Running Time for creating topic result file:  0.004113912582397461 seconds
Total Running Time:  35.91333603858948 seconds
|    Topic      |                Terms (Top 5)                          |
|    Topic 1    |  result, investigate, significant, indicate, reduction
|    Topic 2    |  affect, observe, plant, water, difference
|    Topic 3    |  bacterial, surface, bacteria, show, high
|    Topic 4    |  layer, also, system, adhesion, induce
|    Topic 5    |  use, cell, activity, growth, condition
|    Topic 6    |  lead, resistance, significantly, produce, function
|    Topic 7    |  production, control, capsule, test, grow
|    Topic 8    |  form, infection, treatment, may, environment
|    Topic 9    |  biofilm, formation, study, implant, isolate
|    Topic 10   |  strain, protein, specie, different, associate

10/10
```

Figure 4: Topic Modeling

```
PMID:     23845046

Pub_date: 2014-09-09

Tilte:    Microbiological diversity of peri-implantitis biofilm by Sanger sequencing.

Authors:  Magda Feres, Luciene C Figueiredo, Marcelo Faveri, Ennyo S C da Silva, Jami

Abstract:
To examine the microbial diversity associated with implants with or without peri-impl
inical conditions.

Keywords: bacteria, biofilm, peri-implantitis, Microbial diversity, Dental implant

Enter [S]earch another paper | [B]ack to the search page | [M]ain menu
```

Figure 5: View of Literature

```
Topic 1:

affect          observe         plant           water           difference
promote         microscopy              initial         thin            appear
exposure                hydrophobic             heterogeneous           management              catheter
deposition              uv              local           engineer                sensitive
dose            possess         salmonella              potentially             machine
venous          absence         shape           interestingly           certain
lifestyle               complete                processing              toxicity                tobramycin
exist           strongly                short           positively              tooth
ethanol         latter          former          mat             systemic
prolong         raw             anammox         polluted                artificial
quality         success                 take            ozone           shear
conclude                evolution               promising               leave           heat
transcriptional         lactobacillus           optimize                hydrodynamic            angle
virulent                proteobacteria          milk            clearly         indeed
heterotrophic           evolve          pgpr            option          polar
hemolytic               cool            history         explained               surgical
positively_correlate            trend           statistical             scanning                bulk
notably         candidiasis             suppression             vessel          consequently
staphylococci           disinfectant            replace         water_distribution              cow
donor           chlorine                forming         dominance               experimentally
```

Figure 6: View of Topic with terms

- **Conclusions**:

Biomedical research is the huge area of science which include lots of process which relate to biological. It also involves some causes of disease through careful experimentation, observation, laboratory work, analysis, and testing (SUBR, 2019). Most of biomedical researchers sure and search experimentation as literature at literature databases such as PubMed. It is the reason the size of the biomedical literature database keeps increasing every year. Therefore, literature database is really import for all biomedical researchers. A well-design database will be able to accelerate their research progress and discover more knowledges at the biomedical field. Therefore, this project can help biomedical researchers to get their research done easily.

- **Challenges:**

  Data Collection. Sometimes the xml format result will get little problem such as missing data or missing partial data. The reason is that PubMed may not have the information of the literature I have search such as author's first name, last name, or abstract. Those missing data will make my script crash when I read the xml format data. The solution to solve this problem is setting up some rules to avoid that situation, but it might have some hidden problems which my rules cannot deal with.

Second problem of data collection is I was planning to collect more data for my database. However, the database I have been using is at my local server, and it has no enough to contain more data. I tried to use the odin server, but the server doesn't have some python library I need such as mysql pyhton library.

References:

[1] NCBI Resource Coordinators. Database resources of the national center for biotechnology information. Nucleic acids research, 43(Database issue):D6, 2015.

[2] Sybrandt, J., Shtutman, M., & Safro, I. (2017). MOLIERE: Automatic Biomedical Hypothesis Generation System. *KDD : proceedings. International Conference on Knowledge Discovery & Data Mining*, *2017*, 1633–1642. doi:10.1145/3097983.3098057

[3] Abdullah Ayedh, Guanzheng Tan, Khaled Alwesabi, andHamdi Rajeh.The effect of preprocessing on arabic docu-ment categorization.Algorithms, 9(2):27, 2016.

[4] Alper Kursat Uysal and Serkan Gunal. The impact of pre-processing on text classification.Information Processing &Management, 50(1):104–112, 2014.

[5] B. V. Barde and A. M. Bainwad, "An overview of topic modeling methods and tools," *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, 2017, pp. 745-750.

[6] E. S. Negara, D. Triadi and R. Andryani, "Topic Modelling Twitter Data with Latent Dirichlet Allocation Method," *2019 International Conference on Electrical Engineering and Computer Science (ICECOS)*, Batam Island, Indonesia, 2019, pp. 386-390.