A preliminary literature review

1 Motivation

Literature research is one of the most important parts of scientific publications like a bachelor or master thesis, journals, etc. Finding appropriate literature is one of the most time-consuming elements in academia. Also, it is getting to take more time day by day because the number of publications grows exponentially. The aim of this article is collecting preliminary information to develop literature recommendation software.

2 Introduction

Investigating related works gives better vision to see big picture of searched field and allows to know common practices and problems. To this end, I reviewed three papers:

1. Scienstein: A research paper recommender system, [Gipp et al., 2009]
2. A fast content-based recommendation system for scientific publications, [Achakulvisut et al., 2016]
3. Topic modeling driven content based jobs recommendation engine, [Bansal et al., 2017]

Following sections are mostly formed from these papers. I try collect common practices and problems in one place and try to figure out big picture. Also, in the last section, I listed techniques/methods/algorithms etc. anything it is needed to develop a recommendation system for scientific papers.

3 Recommendation System

Developing a recommendation system needs to be experienced on a couple of things like data pre-processing, dimensionality reduction techniques, similarity techniques, and software development skills. Apart from that we need to decide how to use such that software. Most of the commercial recommendation software like Netflix, Spotify, etc. use past activities of the users or get preferences directly from the users to make better recommendation. This type of technique is called as '**Collaborative-based technique**' that needs some input like voting, tagging from users.

There is another type of technique named '**Content based technique**' that does not need an input directly from users. It is able to make recommendation based on only given text data. This technique is more appropriate for literature recommendation because providing user input for scientific papers is also hard and time consuming. Content based technique extracts 'features' from the given text-data that will be the scientific papers in out case. The words in the text-data can directly act as features. In this case, we can use LSA, SVD. However, a better representation of features in the text-data is 'topics'. Topics can be defined as the most significant words used in the text-data. In this case we can use a topic modeling technique, LDA. After generating latent topics, the software make recommendation based on similarity between the latent topics and the topics of given text as depicted in 1.

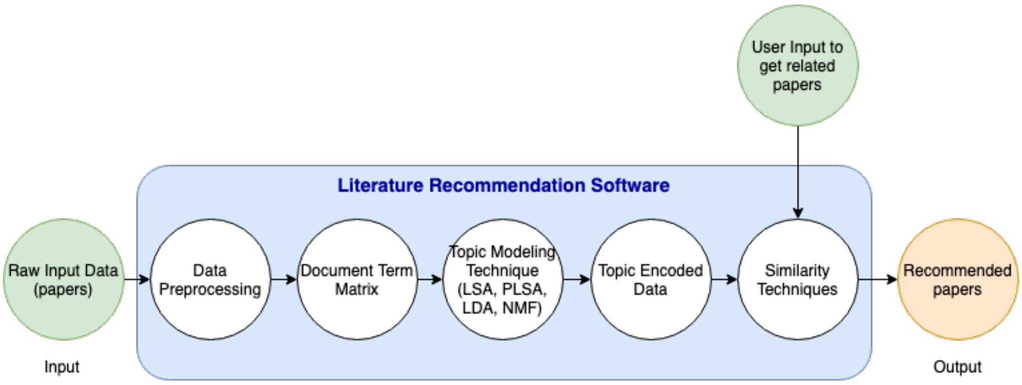

Figure 1: Preliminary architecture of the literature recommendation software

3.1 Data Preprocessing

1. tokenization
2. lemmatization
3. removing stop words

3.2 Topic Modelling Techniques

1. Latent Semantic Analysis (LSA)
   LSA basically applies Singular Value Decomposition over document-term matrix of the input data.

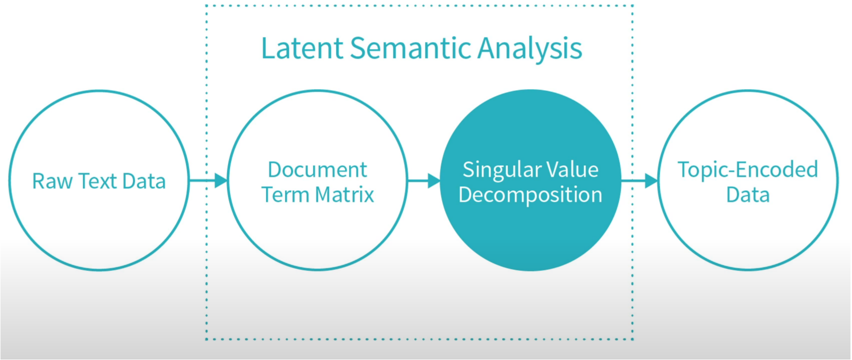

Figure 2: LSA model.(Taken from `https://www.youtube.com/watch?v=YX4xRIQ84ZO`)

2. Latent Dirichlet Allocation (LDA)
   The aim of the LDA is to create representations of the text data in terms of the topic or latent feature. We are going to be able reduce dimensionality of the text data. Under the hood, this model uses **Gibbs Sampling** to assign topics to words correctly.

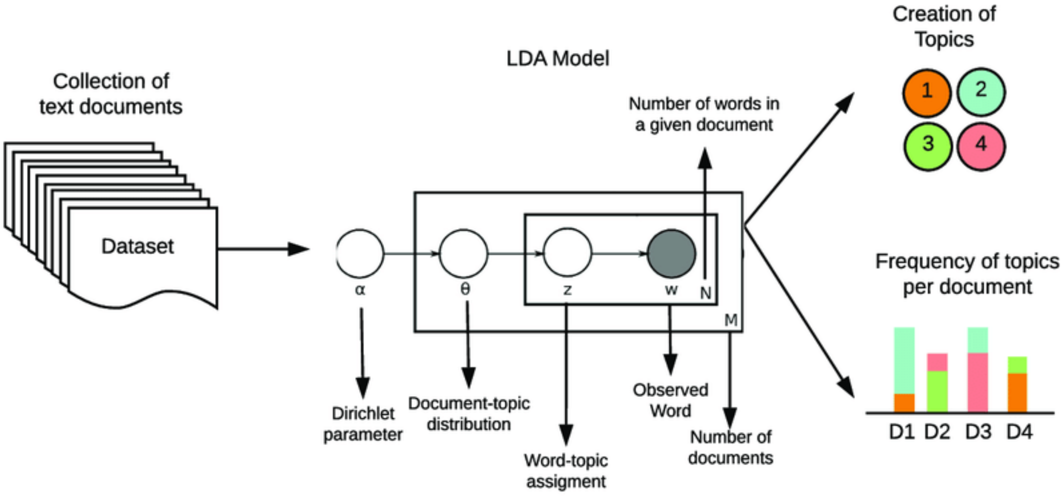

Figure 3: LDA model.(Taken from `https://www.researchgate.net/figure/Schematic-of-LDA-algorithm_fig1_339368709`)

3. Non Negative Matrix Factorization (NMF)
   investigate

3.3 Similarity Techniques

The features are extracting using topic modeling techniques. We need to find similarity between given specific text and the latent feature. To this end, we will compute similarity matrix using one of the following methods below or maybe we can ensemble a couple of them:

1. Flexible string matching
2. cosine similarity measure
3. graph based page rank approach

4 In practice

1. **gensim**: for applying topic modeling. e.g. LDA
2. **nltk**: for part of speech tagging, tokenization
3. **sklearn.feature_extraction**: to create
   (a) document-term matrix(CountVectorizer, TfidfVectorizer)
   (b) remove stop words.
4. **scipy.sparse**: to deal with sparsity

Apply data preprocessing techniques step by step to find best technique for given data. For example, removing all words except nouns or keeping just nouns and adjectives together.

5 Conclusion

I learned theoretical knowledge about LSA and LDA topic modeling techniques. It seems that applying these approach with built-in libraries is not a big deal but tuning parameters will be hard. Also, data preprocessing steps like extracting text from PDF files, removing stop words, tokenization etc. will take much time. However, it will be better to start with data preprocessing to get used to input data. After getting document-term matrix of the input data, we will be able to apply topic modeling techniques.

5.1 Questions?

1. What would be the input of the software? document itself, user voting over document relevance, tagging
2. How to make recommendation? based on similarity matrix?
3. How to measure accuracy of the model?
4. Is the dataset in English?

References

[Achakulvisut et al., 2016] Achakulvisut, T., Acuna, D. E., Ruangrong, T., and Kording, K. (2016). Science concierge: A fast content-based recommendation system for scientific publications. *PLOS ONE*, 11(7):e0158423.

[Bansal et al., 2017] Bansal, S., Srivastava, A., and Arora, A. (2017). Topic modeling driven content based jobs recommendation engine for recruitment industry. *Procedia Computer Science*, 122:865 – 872. 5th International Conference on Information Technology and Quantitative Management, ITQM 2017.

[Gipp et al., 2009] Gipp, B., Beel, J., and Hentschel, C. (2009). Scienstein: A research paper recommender system.