Thesis Recommender System Report

June 4, 2022

1 Thesis Recommender System – Report by Dominik Macko

Our task was to try to improve the recommender system which is used in IS of Masaryk University. Due to the data supplied we were only able to use methods of content-based recommendations.

1.1 Project Experience

In this section I describe my contribution and explain the steps taken in the relevant parts in a brief manner. Overall I would say the project experience was good and I learned many new things.

There were multiple tasks during the project and below I list the ones I fullfilled: - data analysis - implement "statistical" models (keywords baseline, metadata model, BM25 model, ensembles of these models) - plot models' recommendation similarities projections and statistically compare models' recommendations Now I will go through these tasks more in detail.

1.1.1 Data Analysis

I tried to perform some simple analysis of the data needed for our task as can be seen in `pv254_project/notebooks/analysis.ipynb`. Overall, this was a pretty simple and quick task although it was not the best idea to try to do this when sick.

1.1.2 Models Implementation

For the models implementation I wanted to try out "statistical" models in the sense that they don't use any deep learning or word embeddings and so on. Most of the code is in `pv254_project/rssystem.py`.

I tried to stick to clean code and implement it using extendable interfaces, so there is a class ThesisScorerBase that is used to score theses and ThesisRecommenderSystemBase that then processes the results of a scorer and recommends specific theses (not just scores).

First, I implemented the scorer that uses keywords only which should be the baseline (also most likely the implementation used in IS). The implementation was pretty straightforward as I used a simple jaccard distance of the keywords sets for each pair of theses. Additionally I added tokenized topic to the keywords as recommended.

Following that, I implemented scorer that uses metadata. Metadata here refers to Faculty/Supervisor/Reader which are one hot encoded and concatted together. So the vector for each thesis is the concattened list of one hot encoded faculty, supervisor, and reader. Furthermore, more metadata columns could be optionally easily added or some removed – we didn't actually use the faculty one here as it was irrelevant since we had data only for Faculty of Informatics. Then

for the vector similarity we used plain cosine similarity. I also had to code the cosine similarity and one hot encodding myself here.

Another scorer I implemented uses BM25 for the text attributes (english abstract). BM25 is state of the art in lexical (syntactical) Information Retrieval and is quite similar to TF-IDF. But in general BM25 should be better because it deals with term saturation and parametrization of document length. I already used this function instead of tf-idf in Information Retrieval (PV211) course and the results were much better which motivated me to use it here as well. I implemented BM25Plus (BM25+) variant. Even though we used only english abstract it can be easily extended to other columns. For the english abstract to be used I first preprocessed it as:

1. lowercase
2. remove punctuation
3. remove excess white space
4. remove stopwords (here stopwords were tokens that occured in <= 5 or >=len(docs) - 5 documents

Furthermore, to verify that my BM25 is working correctly I compared it to other public implementations on some basic toy texts.

Lastly, I implemented a general scoring ensemble. It works in a way that:

1. Each scorer scores all the theses
2. Scores for each scorer are assigned new score based on the inverse of their rank (1st theses gets most points, the last one gets the least, specifically 1)
3. The scores are summed (multiplication is also available as parameter) for each scorer

The main motivation behind this was to make sure we can aggregate different scorers while not making one overpower the others and to be able to adjust the weights for them. Even though we used weights of 1 for each scorer in the end.

1.1.3 Models Demonstration

In the end we resorted to actually using submodels for evaluation:

- keywords baseline
- keywords+metadata ensemble because metadata alone didn't make much sense
- keywords+metadata+bm25 ensemble where bm25 relied on english abstract

For the visualisation I relied on UMAP but I also looked at TSNE and PCA. I computed the visualisations also for the base BM25 and Metadata models. We briefly tried to inspect and interpret the results but for some models (BM25) it didn't even make much sense except that it most likely showed how the language is changing?

Additionally, I wanted to simply compare how similar the predictions are of our models. For this I took all pairs of models and then calculated mean jaccard score for the recommendations for each thesis. Results were not surprising in the sense that none of the models were really too similar.

1.2 Related Work

In general there is many related work for recommender systems in education, schools, etc. However, I only found two recommender system paper that relates to theses.

The first paper, Recommendation System for Thesis Topics Using Content-based Filtering, concerns recommendation of thesis topics to students based on courses and their grades with the courses. For the text data it uses publication text and course syllabus text. The authors in the paper used basic preprocessing (lowercase, stopwords, lemmatization, ..) and tf-idf with euclidean distance. For evaluation they used 30 students and interviewed them to know the results satisfaction. The approach in how to vectorize the text is similar to ours (even though we used bm25).

The second paper, Examiners Recommendation System at Proposal Seminar of Undergraduate Thesis by Using Content- based Filtering concerns the suggestion of good supervisors for undergraduate theses topics because some supervisors supervise too many students or supervise topics that could be supervised "better" by someone else. For the text data they take abstract + basic theory from the theses supervised by the supervisors and basic theory from thesis proposals (the proposals are from a proposal seminar) They use content based approach with basic text preprocessing (lowercase, stemming, stopwords, ..) and tf-idf followed by k-means clustering with euclidean distance. Then they recommend all lecturers from the cluster sorted based on number of supervised topics. This can be related to us that they actually also used other text data than abstract.

Still, there are many other papers in similar area. For example, recommender systems that recommend university for students based in their interests. Additionally, there are also recommender systems that recommend courses to students for enrollment. However, what I found that can really be related to our work is recommender systems that recommend similar research papers (what mostly concers us are research paper recommenders that don't take users into account)! I will just list some of them (some of them VERY briefly):

- Scienstein: A Research Paper Recommender System – they use hybrid where they use collaborative filtering on ratings and different approaches fo text, references (citation analysis), authors, and sources (this one seems really relevant as they compare two documents of text, still we did not have any data for collaborative filtering as they do)
- A Scalable Hybrid Research Paper Recommender System for Microsoft Academic – another hybrid approach, they combine co-citations, tf-idf weighted vectors of content (same as Pavel tried in our work) and combine them
- A Collaborative Filtering based Recommender System for Suggesting New Trends in Any Domain of Research – collaborative filtering for research papers (not that relevant to us)
- A quality based recommender system to disseminate information in a university digital library – hybrid which uses content-based and collaborative approach
- Research Paper Recommender System for University Students on the E-Book System – they use content based approach but only for titles, they use HCF-IDF