Building a content-based recommender system for scientific papers

1. Introduction

This project is aimed to develop a content-based recommender system for scientific papers using the open source database arXiv downloaded from Kaggle. Text wrangling and pre-processing was performed using natural language tool kit (nltk) library in Python. Two recommendation engines were developed. One is built using basic TF-IDF to convert text into vectors, and another is built using word embeddings produced via TF-IDF Word2Vec algorithm. Both recommendation engines are able to provide top 5 most similar papers to the paper that a user is reading, although user-based data would be needed to evaluate the performance quantitatively. With the help of a recommendation engine, we can efficiently extract relevant information from numerous scientific papers automatically, which would be of great public interest, especially in the scientific community.

1.1. Problem identification and impact statement

arXiv (arXiv 2020) is an open-access repository of electronic preprints (i.e. e-prints) of scientific papers, which covers various fields including mathematics, physics, astronomy, computer science, quantitative biology, statistics, and quantitative finance. It is a collaboratively funded, community-supported resource that is maintained and operated by Cornell University. Since its establishment by Paul Ginsparhg in 1991, arXiv has served the public and research communities by enabling scientists worldwide to share and access research before it is formally published. As of today, arXiv has hosted over 1.7 million scholarly articles. With the exponentially increasing numbers of articles, extracting relevant information efficiently becomes a challenge. Fortunately, it is possible to make the exploration of scholarly articles faster and more accurate, with the help of recommender systems. So far, most recommender systems are available for applications in movie, news, music, and product recommendations. In contrast, relatively fewer efforts have addressed recommendation of scientific papers.

Therefore, the goal of this project is to develop a recommender system that can present the users with related articles based on the article they are reading

1.2. Dataset description

The dataset is downloaded from Kaggle (i.e. arXiv dataset). A description of each of the 10 attributes is provided in Table 1.

Table 1. Description of dataset

No.	Variable name	Description
1	id	ArXiv ID
2	submitter	Who submitted the paper
3	authors	Authors of the paper
4	title	Title of the paper
5	comments	Additional info, such as number of pages and figures
6	Journal-ref	Information about the journal the paper was published in
7	doi	Digital object identifier
8	abstract	The abstract of the paper
9	categories	Categories / tags in the ArXiv system
10	versions	A version history

1.3. Methodology

A general workflow is showing in Fig. 1. In brief, the original database downloaded from Kaggle (https://www.kaggle.com/Cornell-University/arxiv) was loaded to the python jupyter notebook. The relevant columns were selected, and column names were appropriately defined. In addition, we cleaned the raw data by dealing with missing values, and duplicated entries. The cleaned text were appropriately pre-processed for subsequent analyses, which included the general steps below: 1) remove unwanted characters such as numbers, and punctuation; 2) convert all characters into lowercase; 3) convert text into individual words (i.e. tokens) using nltk; 4) remove stopwords that are too common and do not qualify for being good keywords for search; 5) convert each word into its lemma word; 5) remove the words with length less than 2; 6) convert the list of tokens to string; and 7) export the cleaned text for subsequent analyses. To better understand the distribution of word counts of each abstract after text pre-processing, the data was visualized via a histogram.

As no users reading data is currently available, we used content-based recommendation system, which recommends articles to a reader by taking similarity of articles. Two content-based recommendation engines were developed. One was based on the TF-IDF, and the other was based on word embeddings. Both recommendation engines were able to take title as input, and output top five similar articles. TF-IDF (Rajaraman and Ullman, 2011) was used to convert the raw text into vectors, which creates a sparse, high dimensional feature, and does not capture the semantic meaning. In contrast, word embeddings (Mikolov et al., 2013) captures the semantic meaning very well, and creates a dense, low dimensional feature.

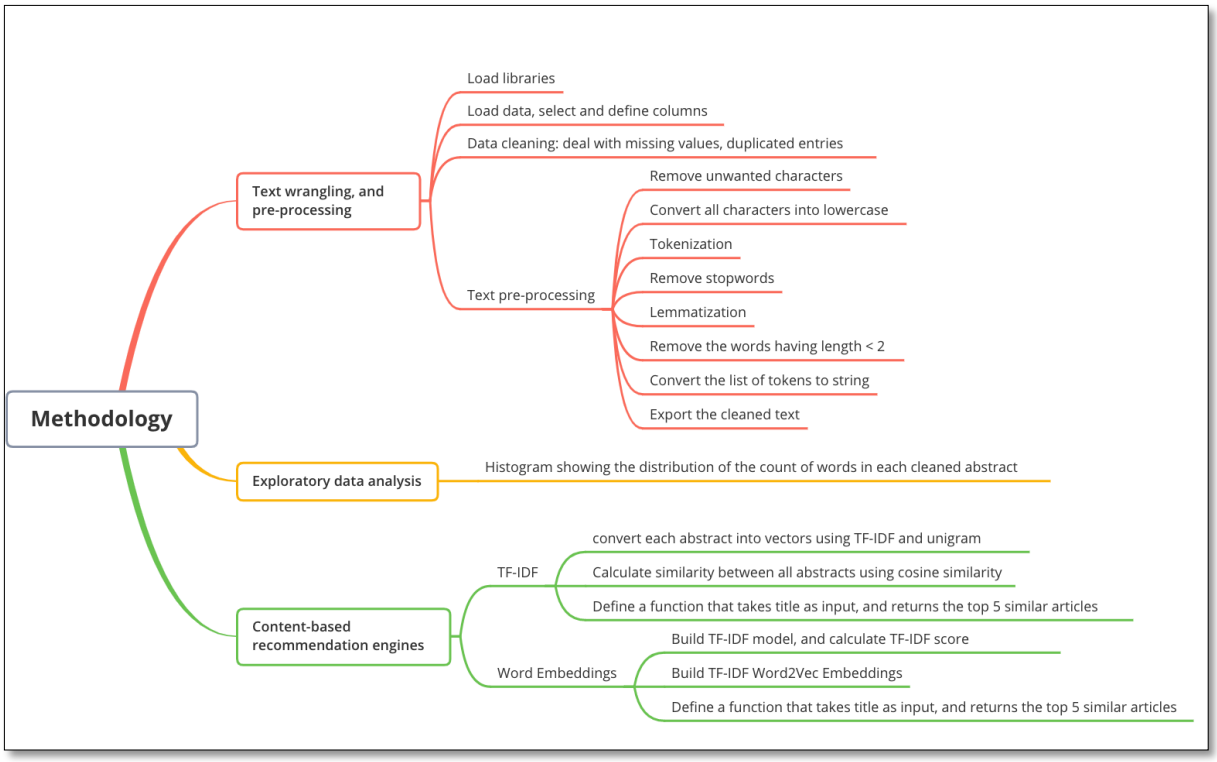

Figure 1. A general workflow.

2. Text wrangling and pre-processing

Articles archived since 2015 were included for this project. Initially, we selected features such as the id, title, abstract, category, and doi of each article. The dataset was load into the jupyter notebook in pandas dataframe format. No duplicated entries were found. There were 4633 out of 18118 articles that do not have doi. Those missing values were represented as 'None' for subsequent analyses. The text in the abstract column were pre-processed via the general steps as specified in the methodology. The cleaned and pre-processed abstracts were stored in a new column named 'clean_abstract'. Figure 2 shows a comparison of an example abstract before and after the pre-processing. The entire process and output is detailed in the jupyter notebook (article_recommend.ipynb).

```
# abstract before pre-processing: abstract
dat['abstract'][0]
```

```
"  We present and study a family of metrics on the space of compact subsets of\n$R^N$ (that we call ``shapes
''). These metrics are ``geometric'', that is, they\nare independent of rotation and translation; and these
metrics enjoy many\ninteresting properties, as, for example, the existence of minimal geodesics. We\nview ou
r space of shapes as a subset of Banach (or Hilbert) manifolds: so we\ncan define a ``tangent manifold'' to
shapes, and (in a very weak form) talk of\na ``Riemannian Geometry'' of shapes. Some of the metrics that we
propose are\ntopologically equivalent to the Hausdorff metric; but at the same time, they\nare more ``regula
r'', since we can hope for a local uniqueness of minimal\ngeodesics.\n  We also study properties of the metr
ics obtained by isometrically identifying\na generic metric space with a subset of a Banach space to obtain
a rigidity\nresult.\n"
```

```
# abstract after pre-processing: clean_abstract
dat['clean_abstract'][0]
```

```
'present study family metrics space compact subsets call shape metrics geometric independent rotation transl
ation metrics enjoy many interest properties example existence minimal geodesics view space shape subset ban
ach hilbert manifold define tangent manifold shape weak form talk riemannian geometry shape metrics propose
topologically equivalent hausdorff metric time regular since hope local uniqueness minimal geodesics also st
udy properties metrics obtain isometrically identify generic metric space subset banach space obtain rigidit
y result'
```

Figure 2. A comparison of an example abstract before and after the pre-processing.

3. Exploratory data analysis

The distribution of word counts of cleaned abstract were visualized in figure 3. It seems the majority of the cleaned abstract have word counts around 50 to 150.

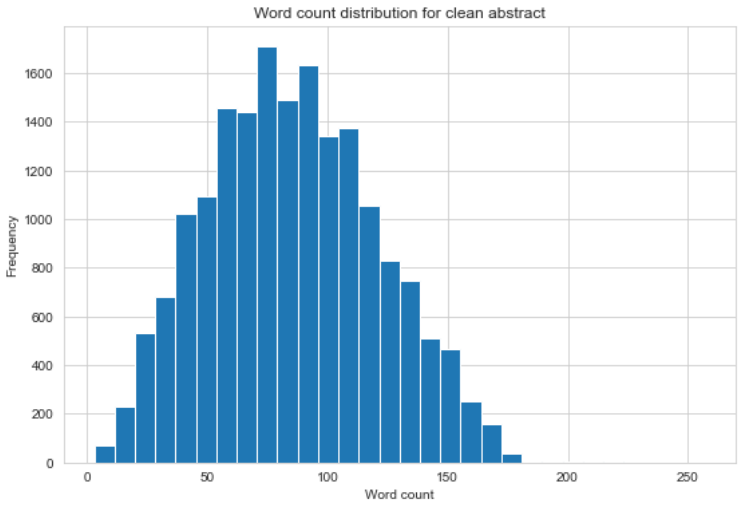

Figure 3. Histograms showing the distribution of numerical variables.

4. Recommendation engines

Two recommendation engines were developed that take title as input, and returns the top 5 similar articles. They were built using TF-IDF, and word embeddings respectively.

4.1. Recommendation engine using TF-IDF

A sparse high-dimensional feature was created via TF-IDF (Fig. 4). Similarities between articles were ranked via cosine similarity, and top 5 similar articles were returned (Fig. 5).

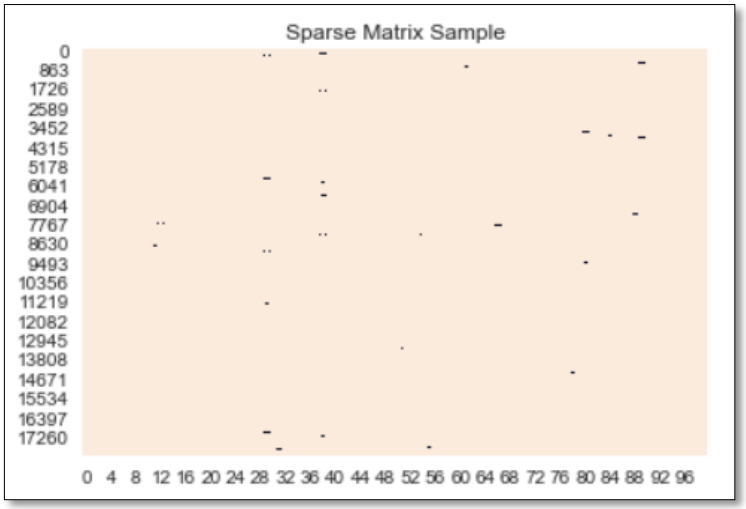

Figure 4. An example of sparse matrix created via TF-IDF.

```
paper_recommend1('Budget Constraints in Prediction Markets')

Here are the top 5 similar articles that may interest you:
--------------------------

#1. Cosine similarity: 0.332
 Title: “Can information be spread as a virus? Viral Marketing as epidemiological
  model”. (DOI:10.1002/mma.3783)

#2. Cosine similarity: 0.321
 Title: “Modeling the residential electricity consumption within a restructured
  power market”. (DOI:None)

#3. Cosine similarity: 0.3
 Title: “Investigating the effect of competitiveness power in estimating the
  average weighted price in electricity market”. (DOI:10.1016/j.tej.2019.106628)

#4. Cosine similarity: 0.286
 Title: “Markets, herding and response to external information”. (DOI:10.1371/journal.pone.0133287)

#5. Cosine similarity: 0.284
 Title: “Improving content marketing processes with the approaches by artificial
  intelligence”. (DOI:None)
```

Figure 5. The top 5 similar articles returned via the recommendation engine built using TF-IDF.

4.2. Recommendation engine using word embeddings

In the above recommendation engine, TF-IDF was used to convert the raw text into vectors, which creates a sparse, high dimensional feature, and does not capture the semantic meaning. In contrast, word embeddings captures the semantic meaning very well, and creates a dense, low dimensional feature.

A dense low-dimensional feature was created via word embeddings produced by TF-IDF Word2Vec that captures the semantic meanings (Fig. 6). Similarities between articles were ranked via cosine similarity, and top 5 similar articles were returned (Fig. 7). More details are provided in the jupyter notebook (article_recommend.ipynb).

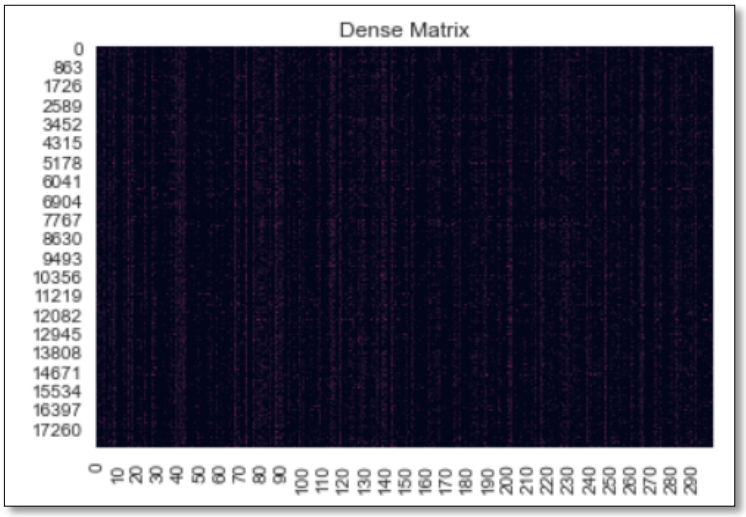

Figure 6. An example of dense matrix created via TF-IDF Word2Vec.

```
paper_recommend2('Budget Constraints in Prediction Markets')

Here are the top 5 similar articles that may interest you:
--------------------------

#1. Cosine similarity: 0.783
 Title: “Modeling the residential electricity consumption within a restructured
  power market”. (DOI:None)

#2. Cosine similarity: 0.776
 Title: “Phase Transition in the S&P Stock Market”. (DOI:10.1007/s11403-015-0160-x)

#3. Cosine similarity: 0.76
 Title: “Investigating the effect of competitiveness power in estimating the
  average weighted price in electricity market”. (DOI:10.1016/j.tej.2019.106628)

#4. Cosine similarity: 0.746
 Title: “Hierarchical structure of stock price fluctuations in financial markets”. (DOI:10.1088/1742-5468/2012/12/P12016)

#5. Cosine similarity: 0.732
 Title: “Satiation in Fisher Markets and Approximation of Nash Social Welfare”. (DOI:None)
```

Figure 7. The top 5 similar articles returned via the recommendation engine built using word embeddings produced via TF-IDF Word2Vec.

5. Conclusions

With the exponentially increasing number of scholarly articles, extracting relevant information efficiently becomes a challenge. Recommendation systems provide a means to make the exploration of scholarly articles faster and more accurately, which would of great importance in terms of information retrieval from flooded scholarly articles.

This project explored the use of basic TF-IDF to build a recommendation engine that takes the title of an article that a user is reading, and returns top 5 related articles that may also interest the user. However, TF-IDF does not capture the semantic meaning, and creates a sparse, high-dimensional feature. Word embeddings create a dense, low-dimensional feature, and captures the semantic meaning quite well. Therefore, we also explored the use of TF-IDF Word2Vec to build a similar recommendation engine.

Both of the two content-based recommendation engines were able to recommend the top 5 related articles to the article that a user was reading. Substantially higher cosine similarity scores were achieved based on word embeddings. This is probably because word embeddings capture the semantic meaning and create a dense, low-dimensional feature, as compared to basic TF-IDF model. Future improvements would include additional features such as authors, categories of articles, and user-based data.

Acknowledgements

This work is only possible with the kaggle open-source database - arXiv. I would like to thank Dipanjan Sarkar, and the Springboard team who provided inputs and guidance throughout this project.

References

arXiv, 2020. arXiv.org e-Print arXiv. URL: https://arxiv.org/ (assessed Oct. 8, 2020)

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., 2013. Distributed Representations of Words and Phrases and their Compositionality. Adv. Neural Inf. Process. Syst. arXiv: 1310.4546

Rajaraman, A., Ullman, J.D., 2011. Data Mining, in: Mining of Massive Datasets. Cambridge University Press, Cambridge, pp. 1–17. https://doi.org/10.1017/CBO9781139058452.002