Building a content-based recommender system for scientific papers

Capstone Project – Springboard Data Science Career Track

Hehuan Liao

Oct. 8, 2020

Problem Identification

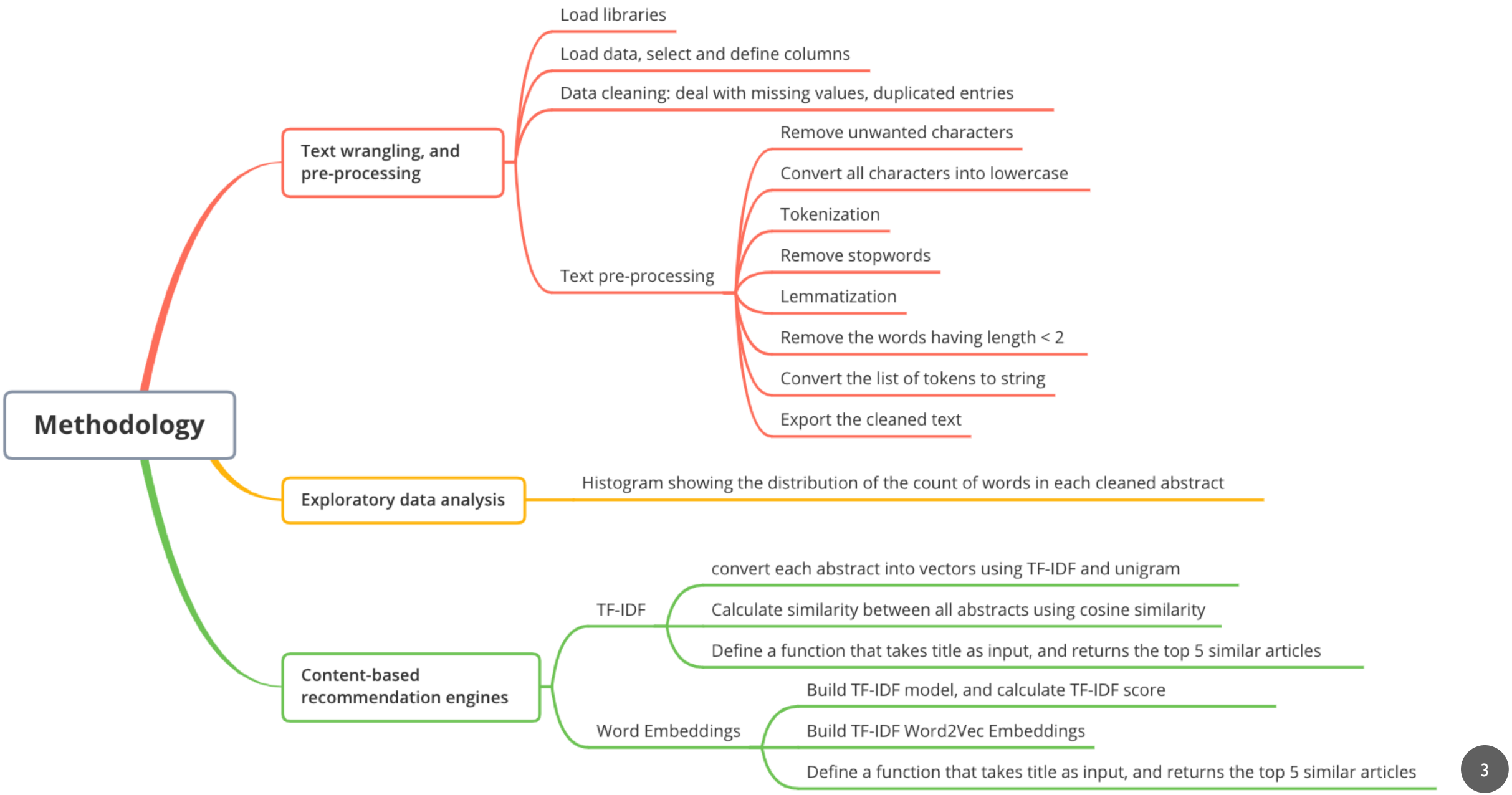
- ArXiv is an open-access repository of e-prints of scientific papers covering various fields
- As of today, arXiv has hosted over 1.7 million scholarly articles, which is increasing dramatically
- Extracting relevant information efficiently becomes a challenge
- A recommender system would make the exploration of scholarly articles faster and more accurate

👉 Overall goal: develop a recommender system that can present the users with related articles based on the article they are reading.

Dataset description

- Data source: arXiv dataset from Kaggle
- Over 1.7 million observations of 10 variables

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


Methodology
Text wrangling, and pre-processing
Load libraries
Load data, select and define columns
Data cleaning: deal with missing values, duplicated entries
Text pre-processing
Remove unwanted characters
Convert all characters into lowercase
Tokenization
Remove stopwords
Lemmatization
Remove the words having length < 2
Convert the list of tokens to string
Export the cleaned text
Exploratory data analysis
Histogram showing the distribution of the count of words in each cleaned abstract
Content-based recommendation engines
TF-IDF
convert each abstract into vectors using TF-IDF and unigram
Calculate similarity between all abstracts using cosine similarity
Define a function that takes title as input, and returns the top 5 similar articles
Word Embeddings
Build TF-IDF model, and calculate TF-IDF score
Build TF-IDF Word2Vec Embeddings
Define a function that takes title as input, and returns the top 5 similar articles

Text wrangling and pre-processing

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

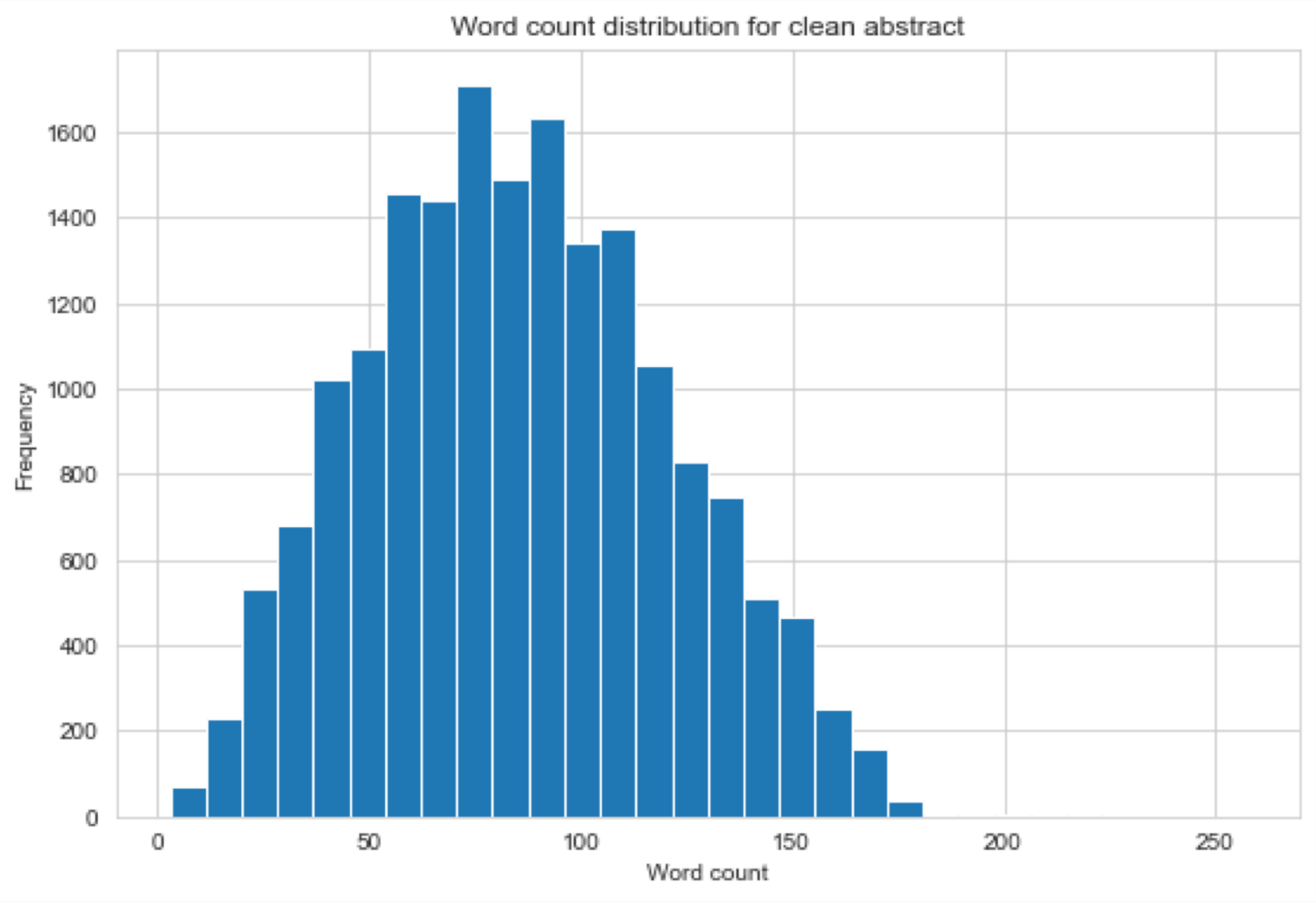
```
"  We present and study a family of metrics on the space of compact subsets of\n$R^N$ (that we call ``shapes
''). These metrics are ``geometric'', that is, they\nare independent of rotation and translation; and these
metrics enjoy many\ninteresting properties, as, for example, the existence of minimal 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'
```

A comparison of an example abstract before and after preprocessing

Exploratory data analysis (EDA)



count	18118.000000
mean	85.373938
std	35.237470
min	3.000000
25%	59.000000
50%	84.000000
75%	110.000000
max	257.000000

Content-based recommendation engines

- TF-IDF (term frequency – inverse document frequency)
  - quantify a word in documents that is weighted to signify its importance in the document and corpus
  - TF-IDF = TF(w)*IDF(w)
    TF(w) = (# of times term w appears in a document) / (total # of terms in the document)
    IDF(w) = log_e(Total # of documents / # of documents with term w in it)

- Word embeddings produced via Word2Vec algorithm:
  - Word embedding is dense representation of words in the form of numeric vectors
  - Word2vec algorithm uses a neural network model to learn word associations from a large corpus of text

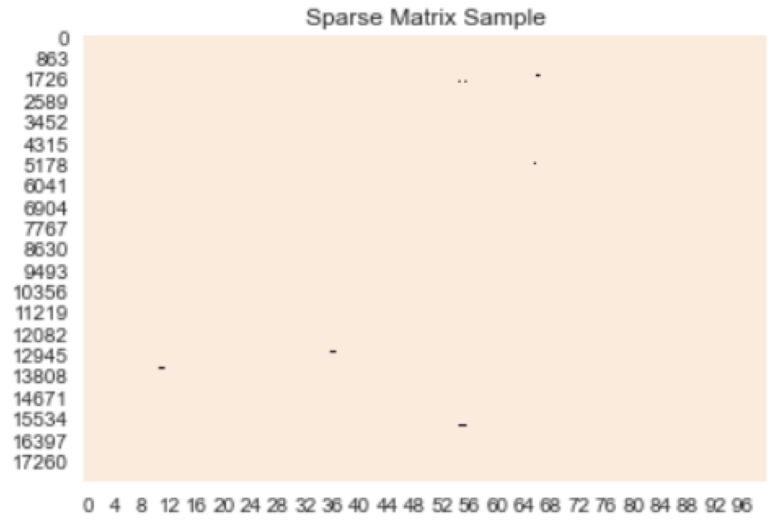
Recommendation engine: TF-IDF

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

#convert the abstract into vectors and use unigram
tfidf = TfidfVectorizer()
tfidf_matrix = tfidf.fit_transform(dat['clean_abstract'])
df1 = pd.DataFrame(tfidf_matrix.todense(),columns=tfidf.get_feature_names())
df1.shape
```

```
(18118, 37948)
```

```
#plot examples of tfidf_matrix
sns.heatmap(tfidf_matrix.todense()
            [:,np.random.randint(0,10000,100)]==0, vmin=0, vmax=1,
             cbar=False).set_title('Sparse Matrix Sample')
```

```
Text(0.5, 1.0, 'Sparse Matrix Sample')
```



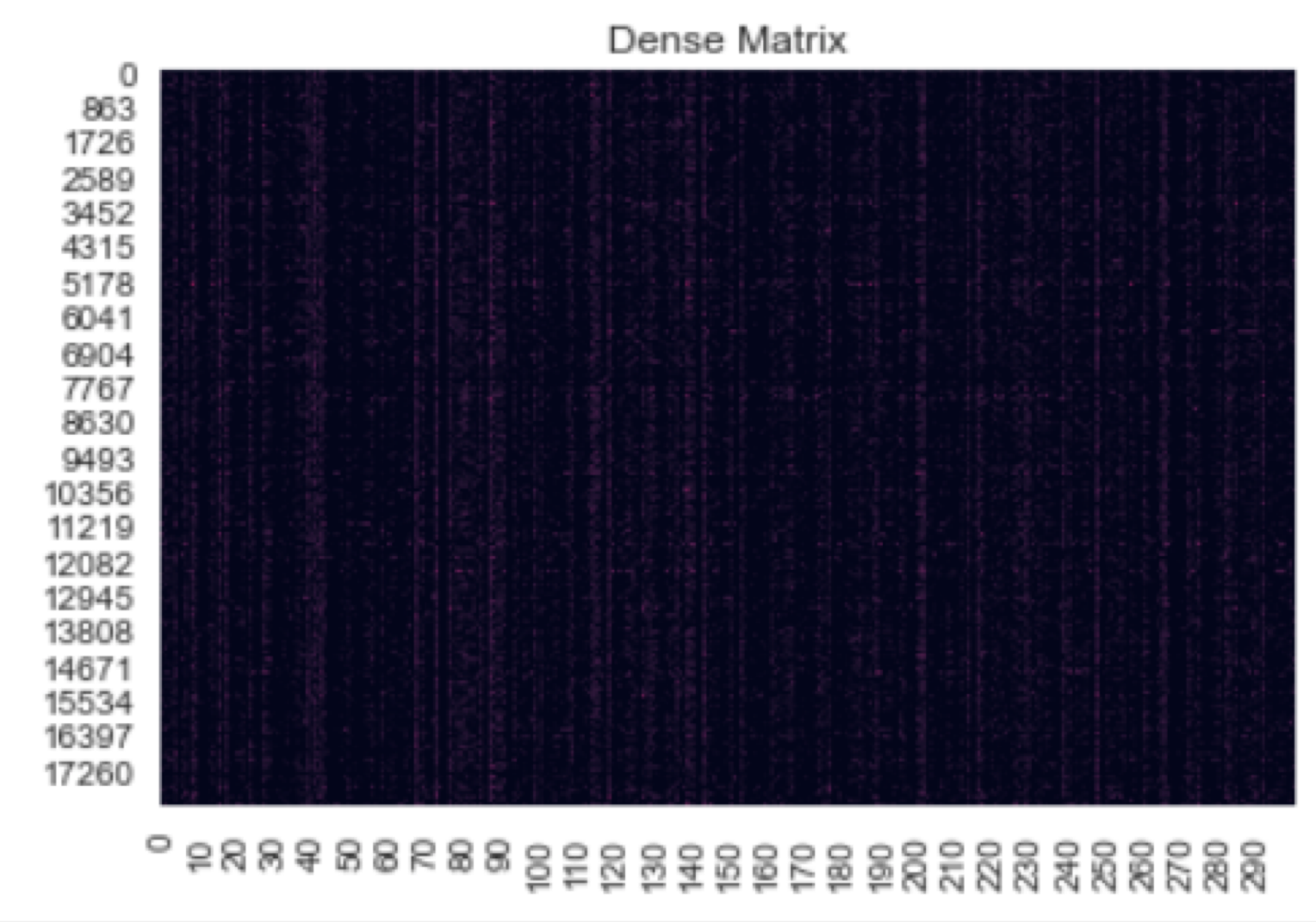
TF-IDF creates a sparse, high-dimensional feature, and does not capture the semantic meaning

Recommendation engine: 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)
```

Recommendation engine: word embeddings



Word2Vec creates a dense, low dimensional feature, and captures the semantic meaning

Recommendation engine: word embeddings

```
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)
```

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.

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 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

- Data source: arXiv database
- Springboard mentor Dipanjan Sarkar, and the Springboard team

The entire project is available:
https://nbviewer.jupyter.org/github/hehuanliao/Springboard/blob/master/Capstone3/article_recommend.ipynb