TOPIC MODELING FOR ECONOMIC NEWS ARTICLES

Springboard Data Science Career Track Capstone Project II

Reuben Yang

INTRODUCTION

- This project deals with the problem of building a machine learning model for news article classification – clustering the articles into a number of groups based on the underlying topic of each article.
- The clients this project serves are the new agencies.
- The dataset used in this project is obtained from the *Data For Everyone Library*[1], which contains 8000 news articles related to economics ranging from 1951 to 2014. The sources of the articles are Wall Street Journal and Washington Post.
- Topic modeling is used to classify the economic articles into different categories and determine the membership of them by discovering the structure in the corpus.

[1]https://www.crowdflower.com/data-for-everyone/

DATA PREPROCESSING

The following steps are performed to preprocess the text data:

1) Remove numbers and special characters and transform all the remaining words to lowercase.

2) Remove superfluous content – some articles starts with author information (e.g. "*Author: Author Name*"), news source (e.g. "The Wall Street Journal Online"), or location information of the news.

3) Lemmatize the tokens to group the inflected forms of a word into a single term. A lemmatizer is used in this case instead of a stemmer since the words after lemmatization remain readable.

4) Only the nouns are kept as they are more useful in revealing the subject of the articles compared with the other forms of words, e.g. verbs and adjectives.

5) The set of English stop words from *nltk.corpus* are filtered out from the list of tokens.

DATA PREPROCESSING (CONT.)

6) The 50 most frequent words in the collection of these articles are summarized and reviewed by me and the following list of words are excluded as they are considered superfluous in delivering the subjects:

   ['*day*', '*today*', '*yesterday*', '*week*', '*month*', '*quarter*', '*time*', '*percent*', '*rate*', '*point*', '*economy*', '*economist*', '*growth*', '*increase*', '*york*', '*report*', '*analyst*', '*term*', '*people*']

7) Eventually, rare words that appear in less than 5 articles and frequent words that appear in more than 50% of the articles are filtered out.

5293 words remain in the resulting dictionary

METHODOLOGY

- Latent Dirichlet Allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora, is applied.
- LDA is a way of discovering topics under which a collection of documents can be grouped and each topic has a list of words associated with it.
- One must figure out what the topics refer to since LDA only provides lists of words that implicitly define the topics but does not automatically label them explicitly.
- Topic Coherence, used to evaluate topic models, has been recently studied by some researchers[1]. Palmetto[2] provides high-level descriptions for popular coherence measures.

[1]M. Röde, A. Both, and A. Hinneburg. Exploring the Space of Topic Coherence Measures. WSDM'15, pages 399-408, 2015.

[2]http://palmetto.aksw.org/palmetto-webapp/

METHODOLOGY (CONT.)

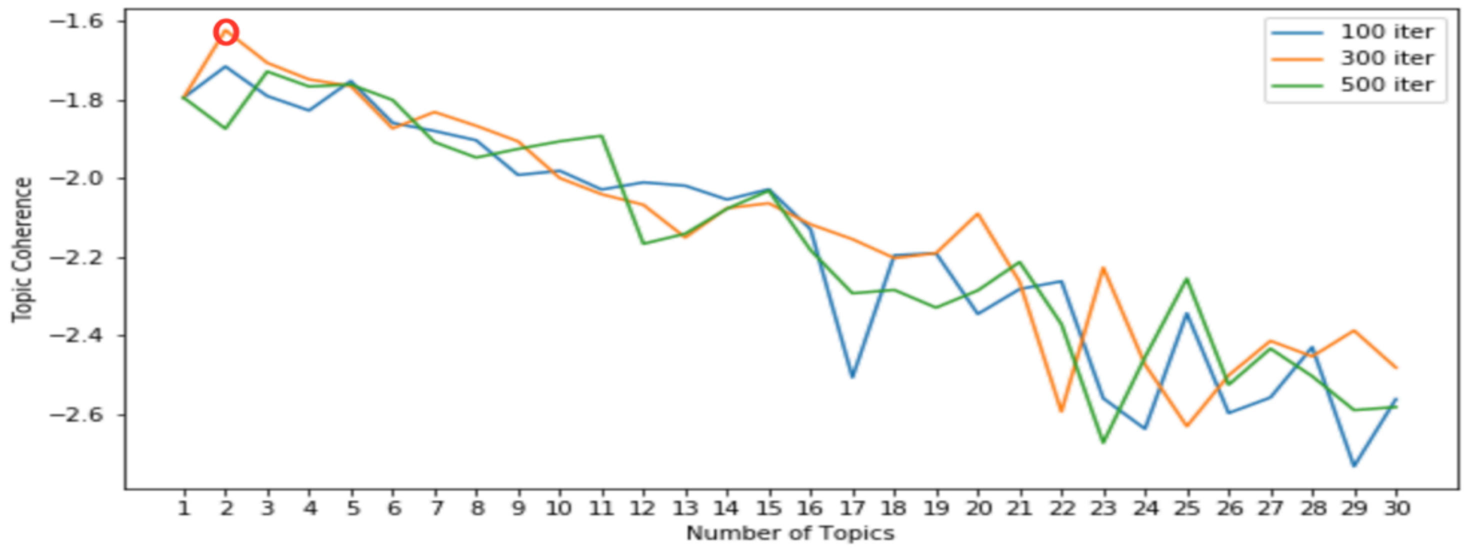
Coherence measures C_v and C_{Umass} are used:

- *Cv* retrieves co-occurrence counts for the given words using a sliding window of size 110. The counts are used to calculated the normalized pointwise mutual information (NPMI) of every top word to every other top word, thus, resulting in a set of vectors – one for every top word.

- C_{Umass} assumes that the probability of a top word to occur should be higher if a document already contains a higher order top word of the same topic. For every word, the logarithm of its conditional probability is calculated using every other top word that has a higher order in the ranking of top words as condition. The probabilities are derived using document co-occurrence counts.

RESULTS

C_{Umass} Model



- The maximum C_{Umass} score is -1.623
- The optimal parameters are 300 iterations and 2 topics

RESULTS (CONT.)

- Top 10 words:

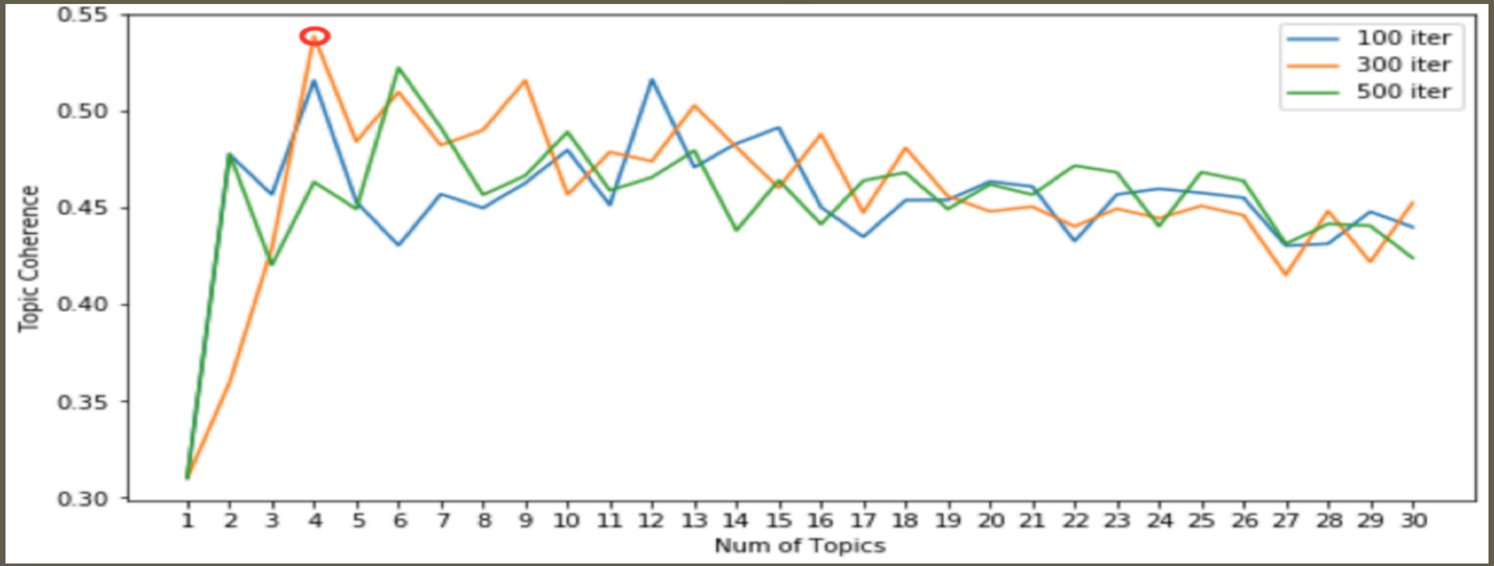
Topic 1		Topic 2	
market stock price interest investor	company bond share dollar index	tax president state government budget	job house deficit bank administration

- # of articles for each topic:

Topic ID	1	2
Count	4363	3637

RESULTS (CONT.)

C_V Model



- The maximum C_V score is 0.538
- The optimal parameters are 300 iterations and 4 topics

RESULTS (CONT.)

- Top 10 words:

Topic 1	Topic 2	Topic 3	Topic 4
tax	stock	company	bank
state	market	sale	interest
president	price	business	reserve
job	index	share	fed
budget	investor	industry	fund
government	dollar	firm	loan
deficit	trading	home	market
unemployment	dow	corp	money
house	bond	service	treasury
administration	share	inc	mortgage

- # of articles for each topic:

Topic ID	1	2	3	4
Count	2573	2689	1358	1380

RESULTS (CONT.)

Evaluation:

1) Randomly select one article from the dataset.

2) Read the picked article and list the 10 keywords from my own perspective. Call this set as H.

3) Classify the article using the *Cv* model and denote the list of 10 keywords as A.

4) Classify the article using the C_{Umass} model and denote the list of 10 keywords as B.

5) Compute *rankA* as the size of the intersection between H and A, and compute *rankB* as the size of the intersection between H and B.

6) The better model (either *Cv* or C_{Umass}) is the one with rank equals to max(*rankA*, *rankB*), i.e. the one has a larger overlap of keywords with H.

RESULTS (CONT.)

- The article with ID 2587 was selected from the dataset.
- Classified by the *Cv* model as its second topic which shares 7 common words with my list, i.e. rank*A* = 7.
- Classified by the C_{Umass} model as its first topic which shares 8 common words with my list, i.e. *rankB* = 8.
- Since *rankB* > *rankA*, the C_{Umass} model is better than the *Cv* model based on this experiment.

My keywords	***Cv* (Topic 2)**	***CUmass* (Topic 1)**
federal	stock	market
reserve	market	stock
bond	price	price
price	index	interest
stock	investor	investor
market	dollar	company
dollar	trading	bond
interest	dow	share
inflation	bond	dollar
share	share	index

CONCLUSION

- Based on the one-article experiment, both topic models built in this project provide article characterizations based on extracted keywords that have nonempty intersections with keywords extracted by one human reader, and hence, the two topics models show a positive promise of potentially good performance in the context of the practical business problem.
- The C_{Umass} model achieves a better performance in classifying the random selected article (Article #2587) as it finds more common words with the keywords extracted by human.
- Since the numbers of the topics (2 and 4) from the two models are significantly smaller than the number of possible words, the dimensionality of the articles is reduced materially.

RECOMMENDATIONS FOR CLIENTS

- The preliminary results show that LDA, combined with coherence metrics C_{Umass} and C_v, indeed produces an implicit classification of a given set of articles.
- The target clients, e.g. Wall Street Journal and Washington Post, can consider using the data preprocessing steps and models presented in this project to reduce the complexity of a large set of articles, through a significantly smaller set of abstract topics with their associated keywords.
- However, the actual effectiveness of this approach with respect to human interpretability remains to be systematically tested.

FUTURE WORK

- The evaluation process could be implemented more comprehensively by asking multiple people reading various articles in the dataset and comparing the lists of keywords with the words from the topics produced by the models. This will strengthen the robustness of the evaluation process as it aims to reduce the bias.
- There are other measures of topic coherence that could be used to train and evaluate the topic models, e.g. *Cp*, *UCI*, *NPMI*, etc.
- Use n-grams so that the order of words and phrases can help capture the articles' topics.
- Consider applying other algorithms in topic modeling, e.g. Latent Semantic Indexing (LSI) and Hierarchical Dirichlet Process (HDP).