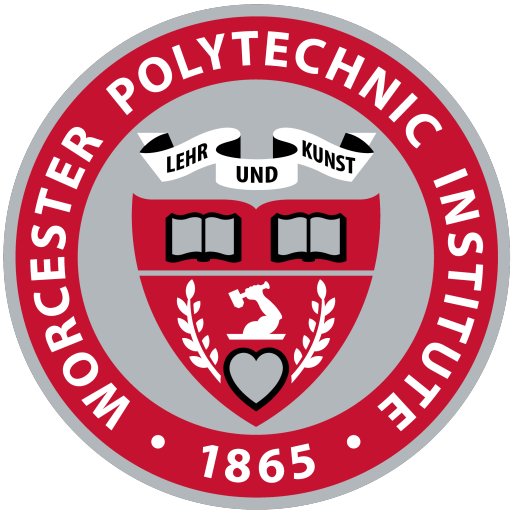
Worcester Polytechnic Institute
FACULTY OF ELECTRICAL AND COMPUTER ENGINEERING



Foundations and Trends in Machine Learning for Engineering

Classifying Reddit Data Using Unorganized Samples vs By Subreddit

Reddit LDA Classifier

Advisor: Prof. Ziming Zhang
Student: Yveder Joseph

WORCESTER MASSACHUSETTS, DECEMBER 2022

Contents

Glossary

Bag-of-words	A bag-of-words is a representation of text that describes the occurrence of words within a document[4].
Corpus	The set of texts used for training[6].
Gensim	Gensim is a free open-source Python library for representing documents as semantic vectors, as efficiently (computer-wise) and painlessly (human-wise) as possible[7].
LDA	Latent Dirichlet Allocation which is an un-supervised, nonparametric and generative method that treats words in documents as if they were generated by probabilistic sampling based on latent variables (topics)[1].
Matplotlib	Matplotlib is a 2D graphics package used for Python for application development, interactive scripting, and publication-quality image generation across user interfaces and operating systems[9].
Sncrape	A python library designed for scraping Reddit, Facebook and Twitter[2].
Stopwords	A set of unimportant words such as "the", "is" and "and"[5].
Tokenize	To split strings into a list of words[6].

UMass calculates how often two words,w_i and w_j appear together in the corpus and it's defined as:

$$C_{UMass}(w_i, w_j) = \log \frac{D(w_i, w_j) + 1}{D(w_i)}$$

where D(w_i, w_j) indicates how many times words w_i and w_j appear together in documents, and $D(w_i)$ is how many time word w_i appeared alone. The greater the number, the better is coherence score. Also, this measure isn't symmetric, which means that $C_{UMass}(w_i,\ w_j)$ is not equal to $C_{UMass}(w_j,\ w_i)$. We calculate the global coherence of the topic as the average pairwise coherence scores on the top N words which describe the topic[8].

1 Previous Applications

Previously LDA has been used to model topics in e-petions. The Obama Administration introduced an electronic petitioning platform called We the People (WtP) to increase government transparency and encourage citizen participation[1]. They provided 39 categories for each petition but the categories were quite broad. Due to the broad scope of the categories you had a situations where 52 percent of the petitions where under the 7 most popular topics Civil Rights and Liberties, Human Rights, Criminal Justice and Law Enforcement, Foreign Policy, Health Care,Family, and Government Reform[1]. WtP-categories where helpful for understanding the federal agencies responsible for specific petitions, but they were not be suitable for understanding issues and concerns petitioners express to the federal government[1]. So due to the limitations of predefined topics Loni Hagen used LDA to generate topics based on the database of WtP petitions.

2 Introduction

This project uses LDA to find the topics within a database of Reddit comments. This database was generated by scraping Reddit using snscrape and then stored in an sqlite SQL database. For this project two different methods were applied and compared. The first method involves taking unorganized samples from the database and creating models for each sample. The second method involves retrieving all the comments from specific subreddits and generating models for said subreddits.

3 Purpose

Reddit is one of the largest social medial networks in the world and by virtue of its structure, where different communities are segmented into different subreddits it contains a rich body of data for topic modeling. I am a frequenter of reddit and I have always been curious on creating a map of the topics being talked about on Reddit.

4 Methodology

4.1 Data Collection

The data collection for this project is a multi staged process. The first step of this process uses the Beautiful Soup library to scrape subreddit names from redditlist.com. The collected names are then suppliet to the snscrape python library which scrapes all the posts and associated comments from said subreddit. The retrieved posts and comments are then stored in an sqlite database based on their respective tables. The comments table contains the fields body(text of of the comment), the date it was created, the id of the comment, the parent id(The id of the comment or post that the comment is responding to). Only the comments were used for this experiment as they were more numerous and contained more text, in total 3.94 million comments were collected.

4.2 Data Processing

The list of comments used to train the LDA models have to be processed, this is done in a two step process. The first step is to tokenize all the comments. After tokenizing any words that are shorter then four letters are removed. For the list of tokens any words that are stop words are removed. The second step is to create a dictionary and a bow corpus. The dictionary contains the words from the tokanized data with its index and count. The dictionary is later filtered so that now words that appear in less then 5 documents, or that appear in more then 50 percent of documents are kept. Furthermore, only the top hundred thousand words are kept. Lastly a bag of words is created for the corpus.

4.3 Model Generation

For generating the models the gensim python libary is used. For each corpus supplied ten models are generated, starting from the first with ten topics to the last with a hundred topics. The UMass score is calculated for each of these models only the best score is kept, for UMass the lower the score the better. The limit for the number of topics was set due to the limitations of my own computer.

4.4 Visualization

For visualizing the results from the models, the Matplotlib library is employed. There are two types of figures generated, the first set of figures is a graph that shows the scores for the ten models generated for each corpus. The second set of figures is a word cloud made up of a sample of ten topics from the best model of a topic.

5 Results

This section contains the results of the models generated by sampling from the database and from the models generated from each subreddit. Not all the subreddits will be shown here as the data was collected from 62 subreddits.

5.1 Sampling Scores

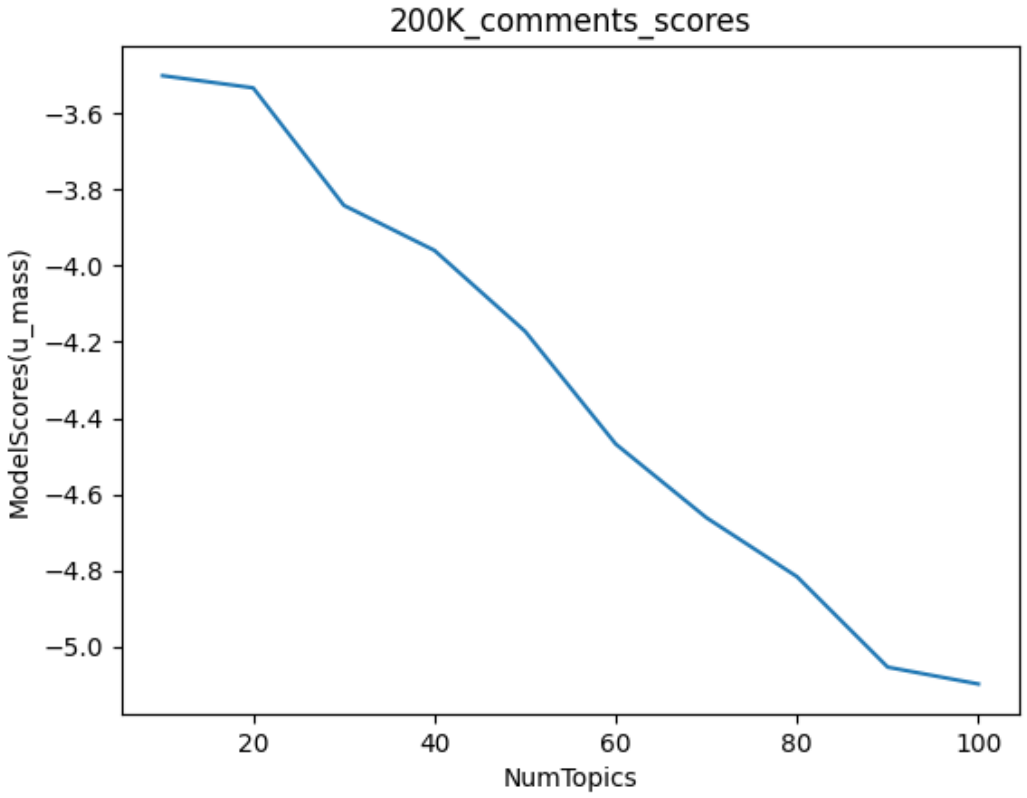


Figure 1: This is the UMass Scores from taking a sample of 200K comments.

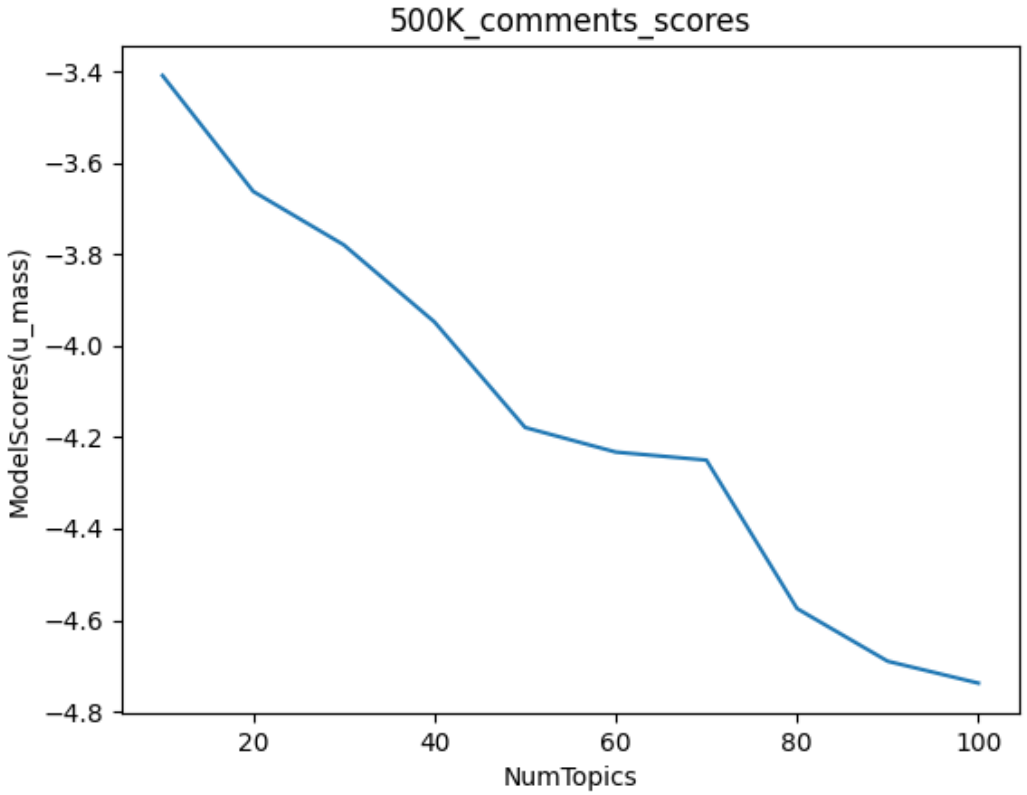


Figure 2: This is the UMass Scores from taking a sample of 500K comments.

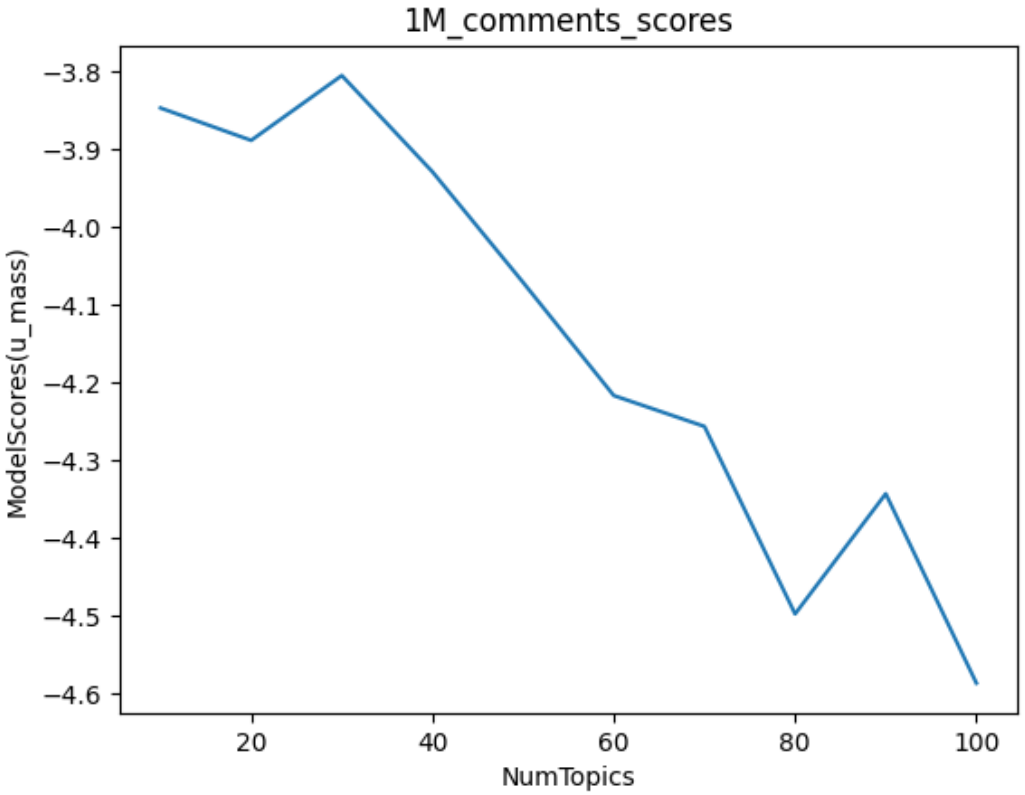


Figure 3: This is the UMass Scores from taking a sample of 1 million comments.

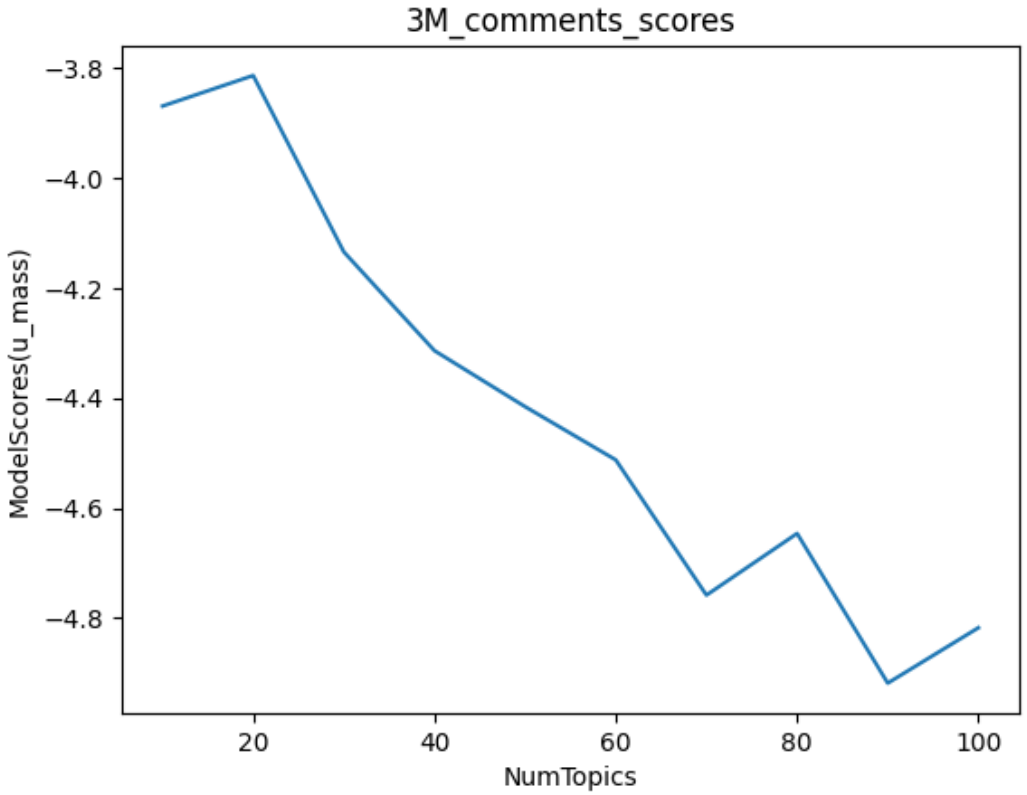


Figure 4: This is the UMass Scores from taking a sample of 3 million comments.

5.2 Sampling Word Clouds

Figure 5: This is the Word Cloud from taking a sample of 200K comments.

Figure 6: This is the Word Cloud from taking a sample of 500K comments.

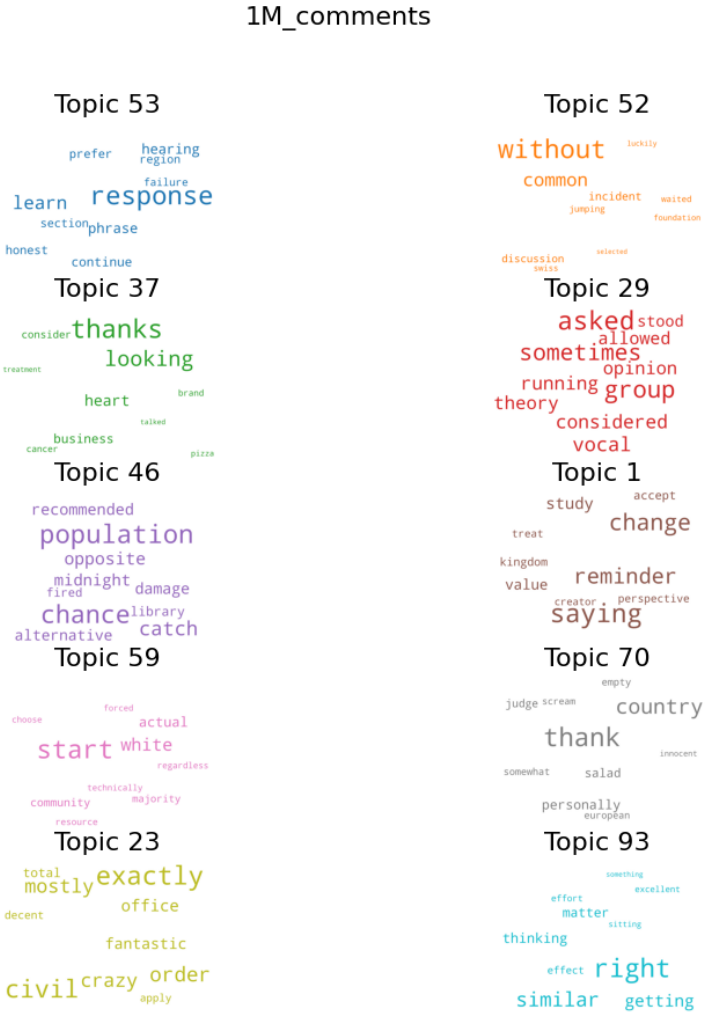


Figure 7: This is the Word Cloud from taking a sample of 1 million comments.

Figure 8: This is the Word Cloud from taking a sample of 3 million comments.

5.3 Subreddit Scores

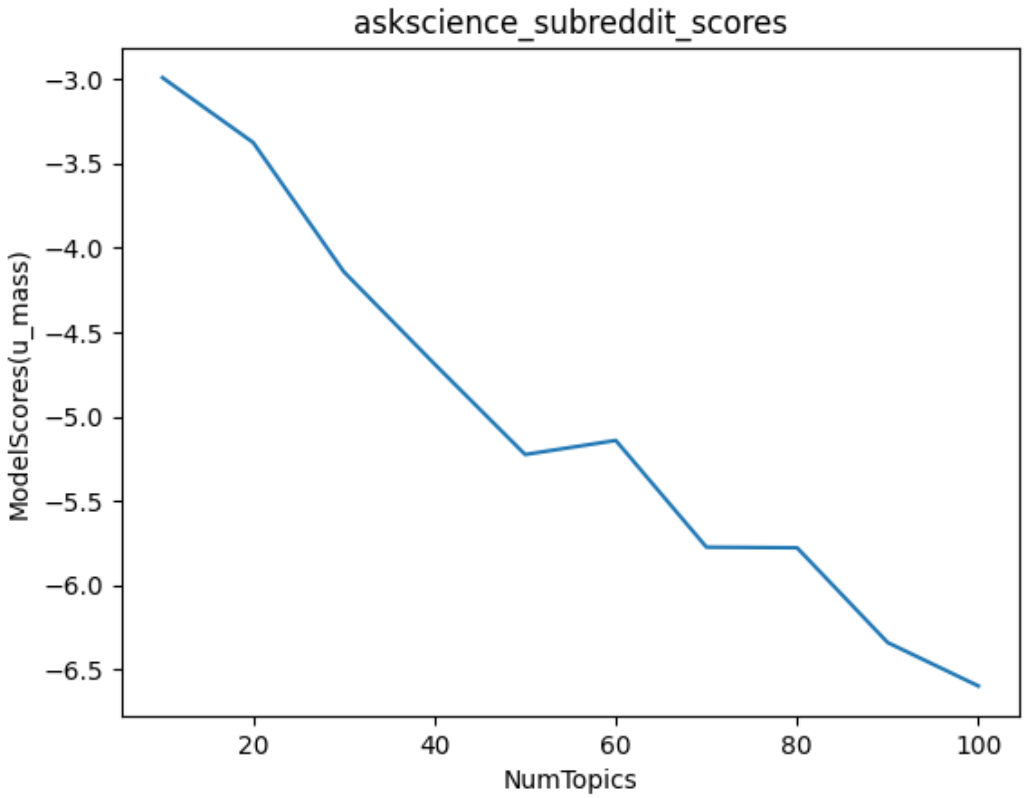


Figure 9: This is the UMass Scores from the askscience subreddit.

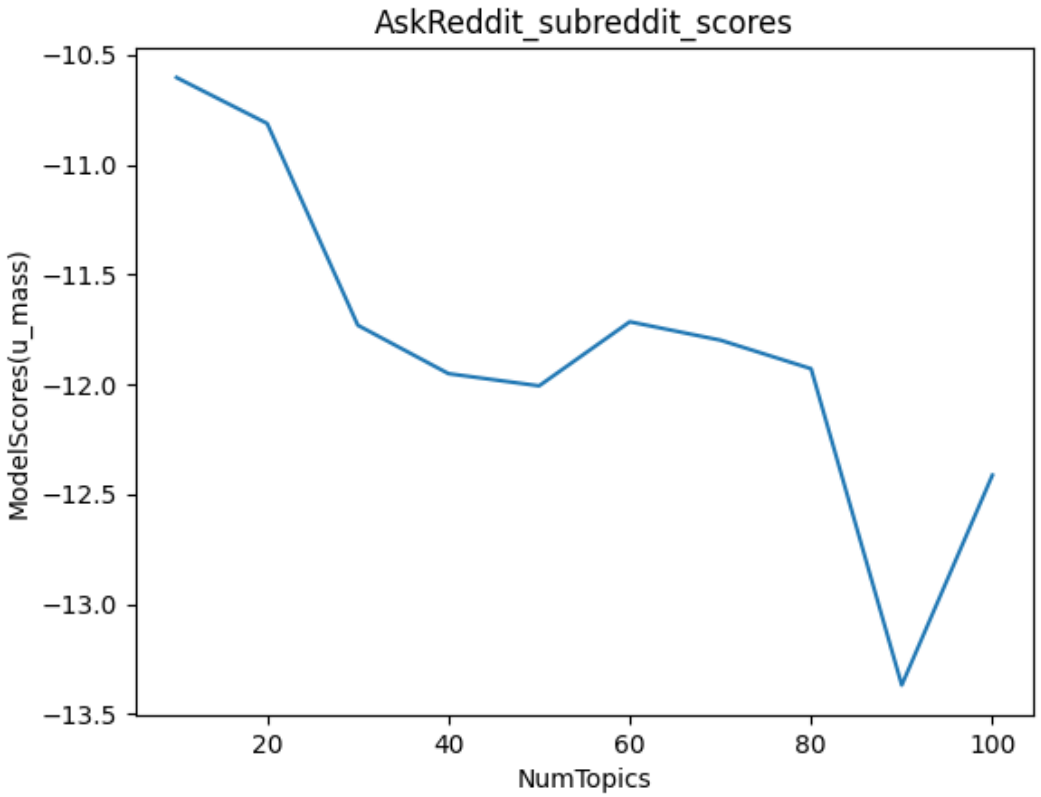


Figure 10: This is the UMass Scores from the AskReddit subreddit.

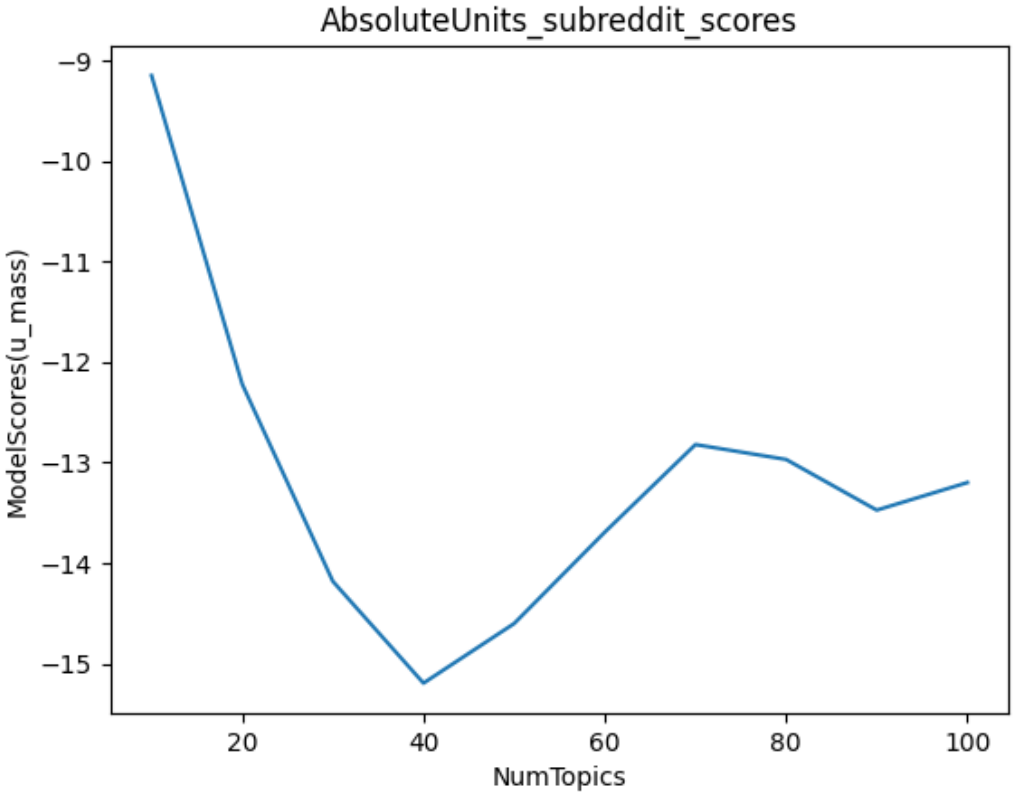


Figure 11: This is the UMass Scores from the AbsoluteUnits subreddit.

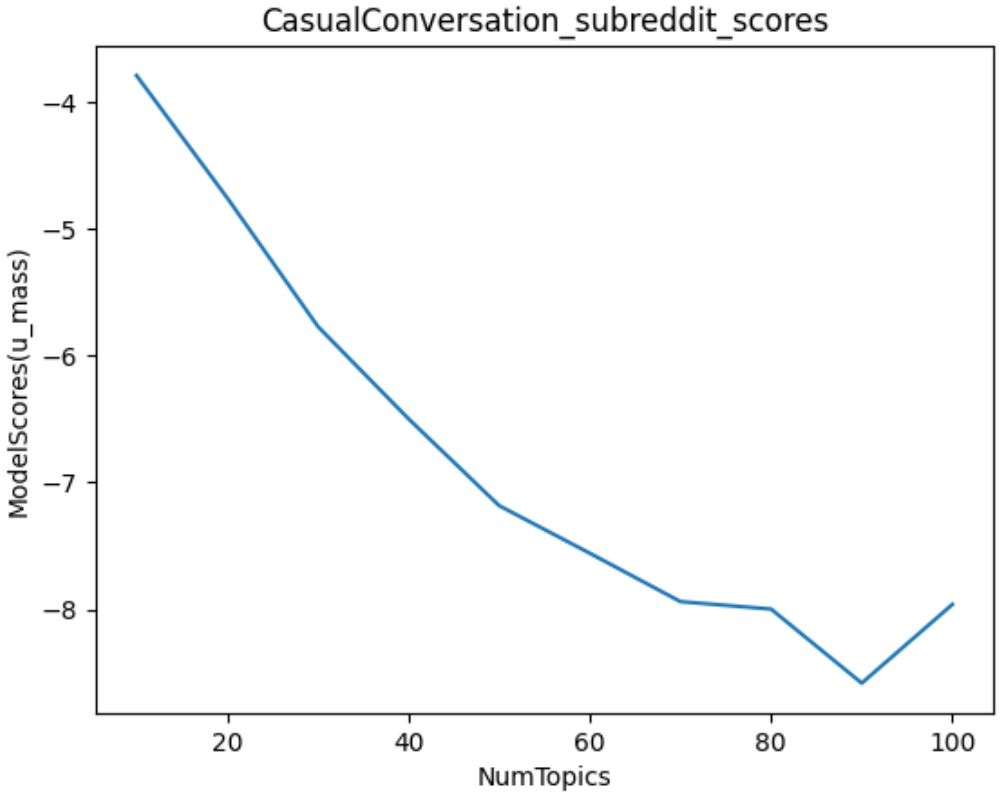


Figure 12: This is the UMass Scores from the CasualConversation subreddit.

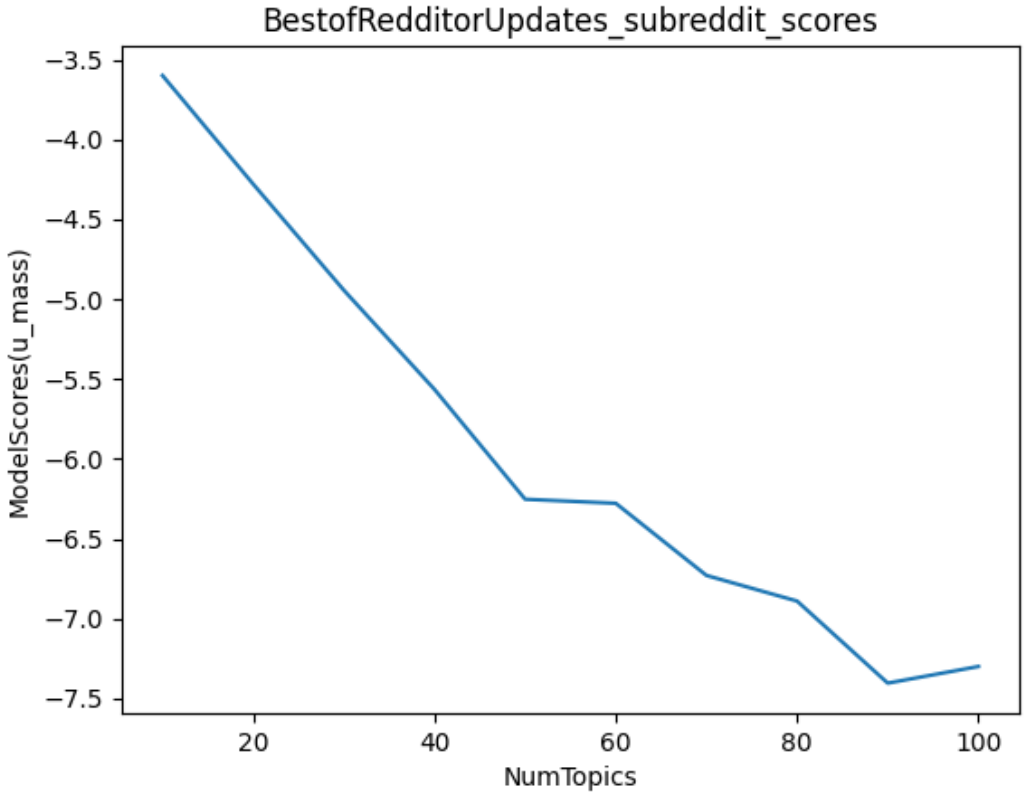


Figure 13: This is the UMass Scores from the BestofRedditorUpdates subreddit.

5.4 Subreddit Word Cloud

Figure 14: This is the Word Cloud from the askscience subreddit.

Figure 15: This is the Word Cloud from the AskReddit subreddit.

Figure 16: This is the Word Cloud from the AbsoluteUnits subreddit.

Figure 17: This is the Word Cloud from the CasualConversation subreddit.

Figure 18: This is the Word Cloud from the BestofRedditorUpdates subreddit.

6 Analysis

From the models generated using the sampled data the scores did not change significantly from each sampling interval. However the same cannot be said for the models generated from the subreddits. The score for these models change by a significant amount between the subreddits. Furthermore, the scores from the models generated for these subreddits is on average significantly larger then those generated from the samples. Additionaly, from inspecting the word clouds, the topics where more clear from the models generated from the subreddits as vs those generated from samples. Lastly for certain subbredits it seemed that the optimal number of topics was reached as in the graph of UMass score vs number of topics a minimum score is reached before the curve begins to increase.

7 Conclusion

Applying LDA to comments on per subreddit basis yielded better results then the models generated from sampling. Better results refers to lower UMass scores and more clear topics in the word cloud. One reason for this maybe due to the computational limitations of only doing up to 100 topics. It could be the case that the texts from a particular subreddit fit into the limited number of topics vs the samples where the texts can from any subreddit and they vary greatly on their subject matters. For future research with access to a more powerful computer it would make sense to increment the number of topics until the UMass scores plateau or starts increasing.

References

[1] L. Hagen, "Content analysis of e-petitions with topic modeling: How to train and evaluate lda models?" *Information Processing Management*, vol. 54, no. 6, pp. 1292–1307, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457317307240

[2] JustAnotherArchivist, "Justanotherarchivist/snscrape: A social networking service scraper in python." [Online]. Available: https://github.com/JustAnotherArchivist/snscrape

[3] "Beautiful soup documentation." [Online]. Available: https://www.crummy.com/software/BeautifulSoup/bs4/doc/

[4] J. Brownlee, "A gentle introduction to the bag-of-words model," Aug 2019. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-bag-words-model/

[5] K. Ganesan, "What are stop words?" Jul 2020. [Online]. Available: https://kavita-ganesan.com/what-are-stop-words/#.Y5Y-zH3MKUl

[6] "Vocabulary - natural language processing with machine learning." [Online]. Available: https://www.educative.io/courses/natural-language-processing-ml/N0Wr9zwpEmv

[7] R. Řehůřek and P. Sojka, "Software Framework for Topic Modelling with Large Corpora," in *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, May 2010, pp. 45–50, http://is.muni.cz/publication/884893/en.

[8] E. Zvornicanin, "When coherence score is good or bad in topic modeling?" Nov 2022. [Online]. Available: https://www.baeldung.com/cs/topic-modeling-coherence-score

[9] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.