Project Report for *RE3: Ukraine War*

Xingwen Xiao
Vrije Universiteit Amsterdam
x3.xiao@student.vu.nl

Dingran Qi
Vrije Universiteit Amsterdam
d.qi@student.vu.nl

Lixiang Zhang
Vrije Universiteit Amsterdam
l.zhang2@student.vu.nl

ABSTRACT

In assignment 2, we finished analyzing data from a Reddit dataset on the topic of the Ukraine War as planned. This report records the general context of the project and what we have achieved and discovered.

1. INTRODUCTION

Reddit is a social news aggregation and discussion platform which is one of the most visited websites in the world.

On Reddit, one of the hot topics is the Ukraine War. Also named Russo-Ukrainian War, it has been ongoing since February 2014, and in February 2022 Russia invaded Ukraine, escalating the war and causing widespread concern in the international community. We will consider February 2014 as the beginning point of the Ukraine War.

In our dataset, we have all comments and submissions data on Reddit since December 2005. Our project involves natural language processing: keyword extraction, sentiment analysis, text classification, word-frequency count and word cloud generation. Our tasks include counting the number of posts and posters overtime; conducting sentiment analysis and political standpoints classification ;displaying the change of political standpoints; generating word clouds, and finding relevant subreddits and the most prolific posters based on political standpoints. Finally, we discuss our results and conclusions.

2. RELATED WORK AND TECHNOLOGY

2.1 Topic Analysis

In this section we review related works that conduct key word extraction and judge if the text is related to the specific topic. Ricardo Campos et. al(2020) [2] present a lightweight unsupervised YAKE! Keyword extraction method from single documents, rests on statistical text features extracted from single documents to select the most relevant keywords of a text. They build evaluation metrics, and compare it against other ten unsupervised approaches and one supervised method to demonstrate the advantages and merits of YAKE! Algorithm. They conclude that YAKE! overwhelmingly outperforms other unsupervised methods in overall effectiveness.

And in another research [3] , they build YAKE! keyword extraction model and compare the extracted keywords against the output produced by the IBM Natural Language Understanding (IBM NLU) and Rake system and get similar conclusion as stated above.

As for other keyword extraction methods, Ammar Ismael Kadhim [7] evaluates the performance of TF-IDF and BM25 on weighting the terms on Twitter. The experiments show that TF-IDF improves the performance evaluation of feature extraction according to the maximum value of the F1-measure is 89.77 for TF-IDF and 89.16 for BM25.

Guo and Xiong [5] propose a keyword extraction algorithm based on TextRank. They build two text networks: one network's nodes are words where the diffusion of two words is defined to calculate the correlation between words and another's nodes are sentences where the BM25 algorithm is used to calculate the correlation between sentences. They then construct a sentence-word matrix to extract the keywords of a text. They apply the algorithm by conducting experiments on the Chinese news corpus.

2.2 Sentiment Analysis

As for the explosion of data in social networks, users express their emotions about events in the network. Many tweets become a database to research sentiment analysis and public affairs.

Before the large-scale application of machine learning, text sentiment classification methods were mainly based on specific words, rules, or semi-supervised [14]. Bo Pang et al.[10] employ naive Bayes, maximum entropy classification, and support vector machines on sentiment classification. However, their method does not outperform the baseline model since there is no way to identify exact sentiment sentences.

Megha et al. [12] combined multiple machine learning methods (support vector machine, decision tree and adaboosted decision tree) to classify tweet sentiment. Tweets are preprocessed and fed into the hybrid model. The final sentiment binary classification results outperform those of the single method.

Alexandros et al.[1] use similar machine-learning technologies to analyse sentiments on tweets. They build a pipeline system on Apache Spark for big data processing. John Snow Labs[9] train a deep learning model to detect emotions into four classes on several datasets. The model is open-source and used on Spark NLP.

2.3 Political Orientation Classification

Political orientation classification is a complex problem, involving multiple directions, such as sentiment analysis, named entity recognition, and text classification. Related

work includes classification (supervised), clustering (unsupervised), and dimensionality reduction methods (unsupervised or semi-supervised). This section discusses classification methods.

Evans et al.[4] study for automated analysis of the ideology of legal texts. They use wordscore and naive bayes for classification and get results higher than the baseline at the time. Yu et al.[15] employ SVM and naive bayes to analyse political speech to classify party tendencies.

Early methods mainly use bag-of-words models. They do not capture contextual effects in general. Word embedding models provide rich contextual representations for political orientation classification. Multi-layer deep learning models bring new ideas to machine learning methods. Rao and Spasojevic[11] combine word embeddings and LSTM to classify multiple languages.

3. RESEARCH QUESTIONS

As per the requirement of the assignment, what we should find out in the exploration are as follows:

1. The count of posts on the topic of the Ukraine War.
2. The count of posters on the topic of the Ukraine War.
3. The distribution of the sentiment in these posts.
4. Generate word clouds about the posts.
5. Some of the most active subreddits and Redditors on the topic of the Ukraine War.
6. What are redditors' political standpoints (pro-Ukraine or pro-Russia) are.

For the first two goals, we may look into the pipeline of topic analysis to find out which posts and posters are talking about the Ukraine War (related work in Section 2.1 and pipeline explained in Section 4.2), and work out the exact count with basic SQL techniques.

For the third question, it is necessary to implement a pipeline of sentiment analysis. With the help of spark-nlp, we built a pipeline described in Section 4.3.

For the last three questions, we classify posts into pro-Russia, pro-Ukraine, or irrelevant as their political standpoints. In this case, we can find out what words are most used by pro-Ukraine Redditors (how to generate word clouds is described in Section 4.5) or pro-Russian Redditors. The classification of politics is a difficult obstacle in our project, the background materials are mentioned in Section 2.3, and the pipeline implementation is mentioned in Section 4.4.

4. PROJECT SETUP

4.1 Data Exploration

The first step to look into the data is to find out the structure of comments and submissions. Spark SQL and the built-in Jupyter notebook provide a shortcut for us.

From the beginning of the war, the total size of comments is approximately 2.0TB, while the total size of submissions is 0.8TB. The size is growing years after years exponentially and highly centralized (prominent long tail effect) in a small number of users and subreddits as we have stated in the plan.

Not all columns (attributes) of data is needed, after selecting specific attributes, typical comments look like:

Attribute	Content
parent_id	t3_1ycrt3
author	Squalor-
body	He won't be happy...
id	cfjjan6
score	11
ups	11
subreddit	pics
subreddit_id	t5_2qh0u
created_utc	The created time in utc.

Table 1: Comment Content

Attribute	Content
author	Some_Crazy_Canuck
title	Horrifying video ...
selftext	empty
subreddit	videos
id	1ydlp2
score	9
created_utc	1392844730

Table 2: Submission Content

Typical submissions look like:

For task of sentiment analysis and political standpoint analysis, title, selftext, body are the most essential attributes, as they indicate what has a redditor post. author, subreddit are also necessary for identifying "Some of the most active subreddits and redditors on the topic of Ukraine War" and "What redditors' political standpoints (pro-Ukraine or pro-Russia) are" in Section 3. created_utc is indispendible because of the time serial chart we intended to create.

Some of these attributes can not be processed directly:

- created_utc is the unix timestamp, which should be converted to the date so that it is human-readable and can be visualized.
- parent_id looks confusing because it is prefixed with label t3 or t5, indicating whether it is the descendant of a submission or comment. It should be taken into account when performing inner join to find out a complete thread on Reddit.
- Many selftext are empty because a lot of submissions do not contain body content. Since some posters treat title as body content, we have to concat the texts in title and selftext to perform subsequent classification.

4.2 Find related Posts

Originally, the dataset contains a huge number of unrelated posts. And we have to filter the submissions and comments related to the Ukraine War. To achieve this goal, we apply two keyword extraction algorithms: YakeKeywordExtraction and BM25 on the posts and compare their performance.

4.2.1 YakeKeyWordExtraction

YakeKeywordExtraction is an unsupervised keyword extraction algorithm. Generally, it involves five steps: (1) text

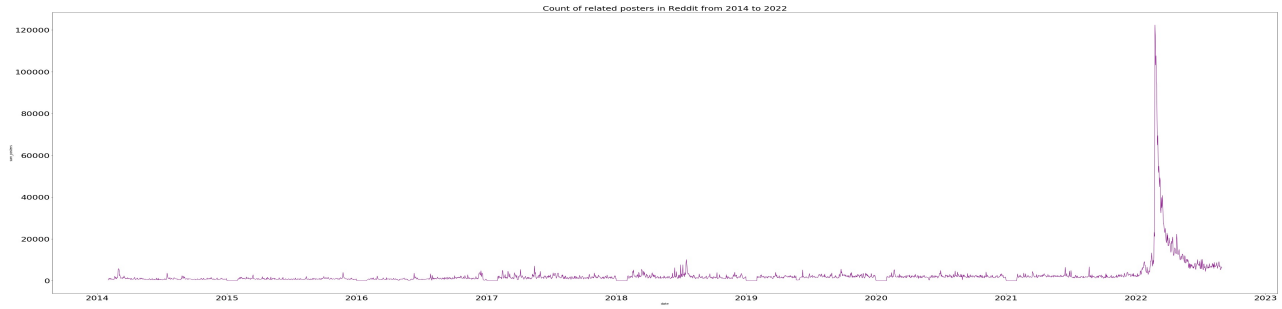

Figure 1: Count of related posters since 2014

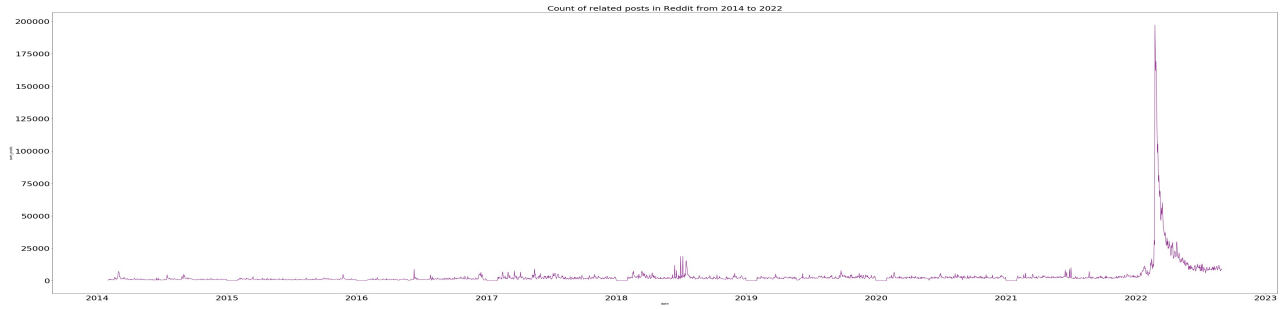

Figure 2: Count of related posts since 2014

pre-processing and candidate term identification; (2) feature extraction; (3) computing term score; (4) n-gram generation and computing candidate keyword score; and (5) data deduplication and ranking.

With the help of sparknlp, we can build a pipeline model to apply the YakeKeywordExtraction algorithm. We set the minimum length of keyword to 1 and the maximum length of keyword to 3. In chronological order, for each post, we extract the top 3 most important keywords.

We make a list with keywords from Russo-Ukraine War Google Trend. It contains nine keywords: "ukraine", "russia","russian","ukraina","putin", "kyiv","moscow", and "vladimir". For each extracted keyword list, we check if it has intersection with the predefined keyword list. If so, we add the post to the filtered files.

After that move, for each month, we join the filtered submission with the comments: the parent id of the comment is the id of submission. We finally get 2,373,520 submissions and 8,666,926 comments remain. We plot the count of related posts and posters overtime, and the number remain steady before February 2022 except for a rise in late 2018, which may caused by Kerch Strait incident. The number of related posts and posters both reach the peak in February 2022 when the Russian invasion of Ukraine began.

4.2.2 BM25

BM25 computes the relevance of a document to a given search query. It is an information retrieval(IR) technique. IR can return relevant information according to the user's needs (query). Essentially, BM25 is an optimisation of TF-IDF.

BM25 is an algorithm rather than a data-driven model. So the query is essential and custom. To extract keywords, we use the dataset from huggingface - russia-ukraine-conflict-articles[8]. This dataset contains 407 New York Times and Guardian reports on the Russian-Ukrainian conflict over a long period (from February to July). The keyword extraction method used is TextRank.

```
['Russian', 'Russian gas', 'Russia's attack', 'Ukraine's forces', 'Russian oil', 'Russian troops', 'Russian artille
ry', 'central Ukraine', 'Ukraine's army', 'Ukraine's military', 'Ukraine War', 'Russian natural gas', 'Russia', 'ea
stern Ukraine', 'Russian missiles', 'Ukraine's armed forces', 'Ukraine president Volodymyr Zelenskiy', 'southern Uk
raine', 'Russian attacks', 'Russia's invasion', 'Russian war crimes', 'Russian forces', 'Russian officials', 'Ukrai
ne', 'northern Ukraine', 'Russian energy', 'east Ukraine']
```

Figure 3: Extracted Keywords for Query

We split the dataset into four long strings. Keyword extraction is performed on each string. And we select the top 50 keywords. The four lists of keywords are filtered to retain keywords that appear more than twice. These keywords are the query of BM25. BM25 will give each given text an associated score. We end up with 27 keywords. Keeping multiple keywords improves the recall score. The keywords are shown in Figure 3.

4.2.3 Comparison

As mentioned in 4.2.2, BM25 requires a list of keywords extracted from a corpus before its evaluation process. As a result, BM25 highly relies on what the algorithm is fed. As Reddit is basically an English site, the corpora we looked for are mainly large media based in the US, like New York Times. This leads to a biased selection result that neglects many pro-Russia Redditors.

The most significant obstacle to implementing BM25 in the pipeline is the ruling out of many relevant comments and submissions while the true positive rate (recall) is too low.

In some circumstances BM25 is useful, but for this project, it is essential to accumulate enough amount of submissions

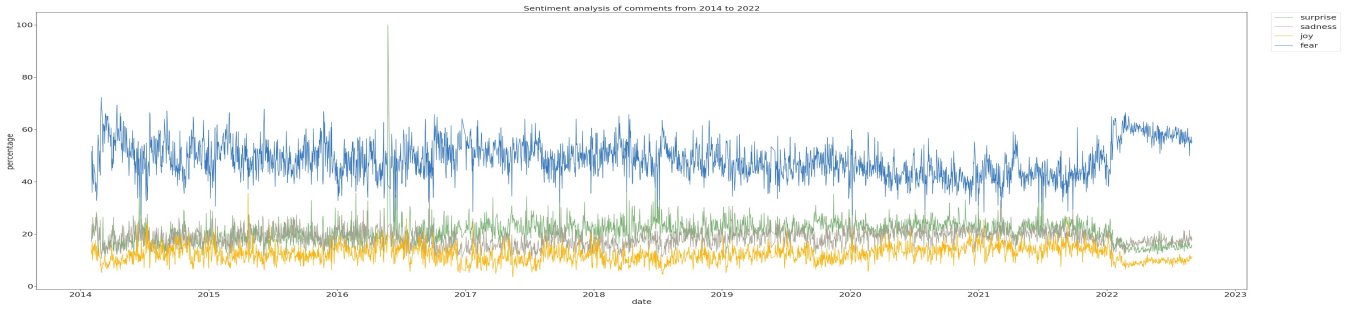

Figure 4: Sentiment of Comments

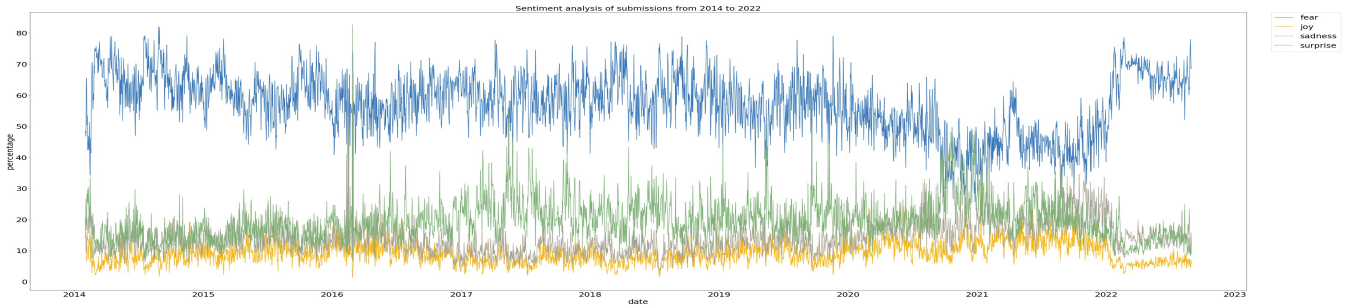

Figure 5: Sentiment of Submissions

and comments on the topic to conduct the following analysis. As a result, we chose Yake as the method of topic analysis.

4.3 Sentiment Analysis

We have gone through a typical sentiment analysis in our experiments. The first step is to transform the data type to the document type which can be processed by spark-nlp. Next, we used the universal encoder to map the sentences to a matrix. With the help of the pre-trained deep learning classifier model, we reached the result of comments sentiment in Figure 4 and submissions sentiment in Figure 5.

The processing pipeline is defined in Figure 6.

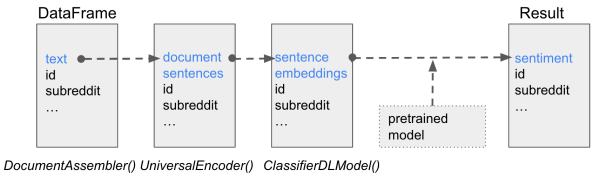

Figure 6: Process pipeline of sentiment analysis

As we can see from the comments sentiments and submissions sentiment, overall, in descending order of quantity, the sentiments are fear, sadness, surprise and joy. And among them, the ratio of fear is overwhelmingly high. At the beginning of the war in 2022, fear also surged to dominate the sentiment of posts.

4.4 Political Standpoint Analysis

We treat political standpoint analysis as a text classification task and construct a pipeline model. The model is a supervised NLP model.

The processing pipeline is defined in Figure 9.

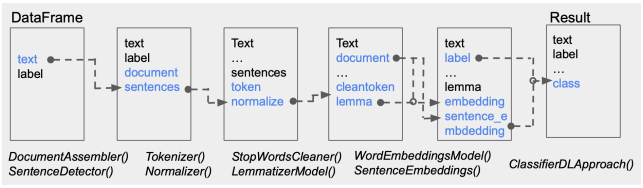

Figure 9: Process pipeline of political standpoint analysis

We sample comments and submissions separately. For comments, we sample 15 samples per month until 2022 and 50 samples per month in 2022. After annotating the comments, we decide to centrally sample the submissions with a deeper understanding of the data. We sample 900 submissions from February, March and April 2014 and 1000 submissions from February and March 2022. We manually annotate the data into three classes (0: ProUkraine, 1: ProRussia, and 2: CantSee).

To balance the data for training, we select 816 data of class CantSee (under-sampling) and reuse the data of class ProRussia (over-sampling). We get a hand-made dataset with 2479 labelled submissions and comments.

Our political standpoint analysis model follows standard NLP preprocessing procedures. They are document assembly, tokenization, lemmatization, stopword removal and word embeddings. The processed data is fed into a deep-learning model for classification. The test results are shown in Figure 10. The accuracy is 76.6%.

We apply our model to all submissions and comments. We find that 15.4% of comments and 54.7% of submissions show a pro-Ukraine standpoint. And 8% of comments and 19.3%

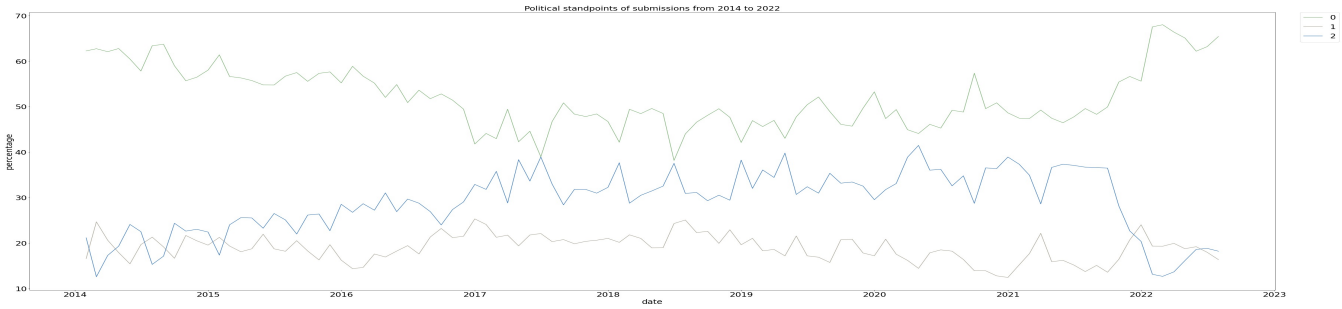

Figure 7: Political standpoint of Submissions

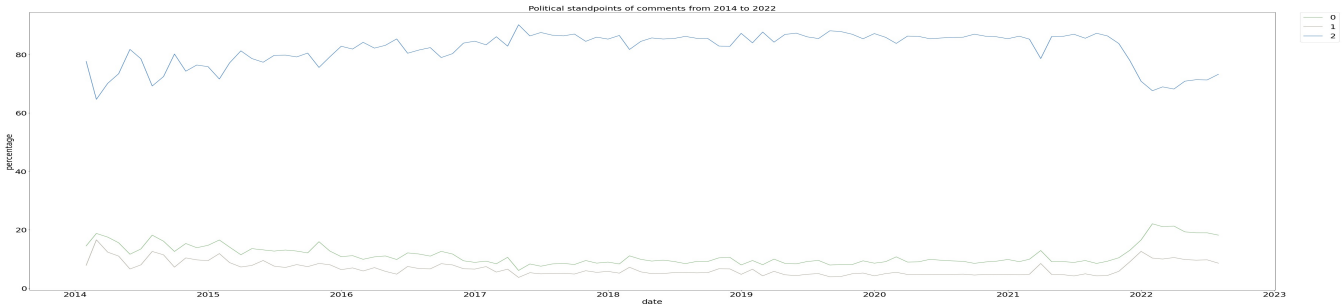

Figure 8: Political standpoint of Comments

of submissions are pro-Russia. We also draw proportional standpoint curve for submissions and comments.In Figure 7, we can see that most people support Ukraine. And after the outbreak of the war in 2022, fewer people are vague about the war, and more people are pro-Ukraine. But in Figure 8, we find that people are less likely to express political leanings in comments or it is difficult to tell it apart with just one comment.

	precision	recall	f1-score	support
0.0	0.61	0.82	0.70	140
1.0	0.87	0.63	0.73	196
2.0	0.85	0.88	0.87	160
accuracy			0.77	496
macro avg	0.78	0.78	0.77	496
weighted avg	0.79	0.77	0.77	496

0.7661290322580645

Figure 10: Test result of political standpoint analysis

4.5 Word cloud

In this section, we generate word clouds based on political standpoints. So the output includes four diagrams: word cloud for pro-Ukraine posts, word cloud for pro-Russia posts, word cloud for pro-Russia posts, and word cloud for pro-Russia posts.

We install nlp package nltk and from nltk we import stopwords. Since the size of comments and submissions data is huge, we sample all the submissions and comments with a ratio of 0.01.

Then for each post, we remove the punctuation marks and unprintable characters, tokenize the text and remove the stopwords, then add it to a list. For each combination of political standpoint and post category, we get a very long word list. We import another package: Counter to get the frequency of each word, select the top 50 most common words and build word cloud with the help of wordcloud package.The word cloud diagrams are Figure 12, Figure 11 , Figure 13 , Figure 14.

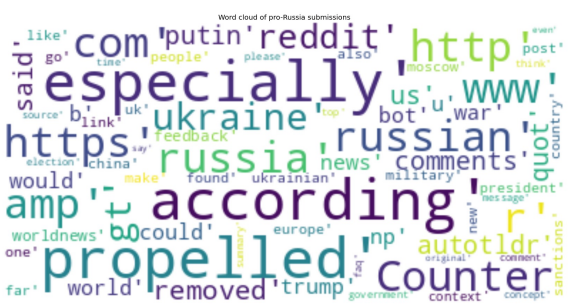

Figure 11: Word cloud of pro-Russia submissions

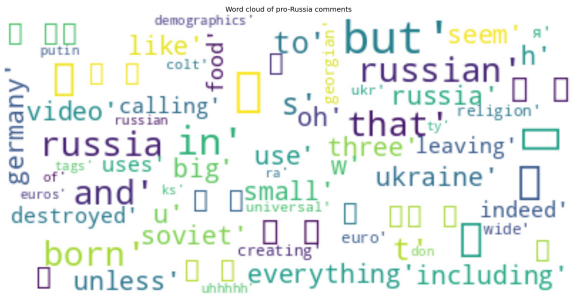

Figure 12: Word cloud of pro-Russia comments

Figure 13: Word cloud of pro-Ukraine comments

Figure 14: Word cloud of pro-Ukraine submissions

4.6 Relevant subreddits and prolific posters

In this section, we analyze the subreddits that appear the most often and the most prolific posters after the political standpoint analysis. For each political standpoint, we union all the comments and submissions data, then separately get the sum of posts grouped by author and subreddit, and select the top 20 subreddits or posters(authors) and draw barplots with the help of seaborn package.The figures are Figure15, Figure16, Figure17,and Figure18 .

For subreddits, the top 3 pro-Ukraine subreddits are worldnews, ukraine and UkrainainConflict, followed by politics and europe—— exactly the same with pro-Russia subreddits. However, for a specific subreddit, the number of pro-Russia posts is approximately half of the number of pro-Ukraine posts. We think that makes sense because in the Internet more people stand on Ukraine's side. And for posters, the top 3 pro-Russia posters are AutoNewsAdmin, AutoNewspaperAdmin, and en, followed by autotldr and rotoreuters. The top 3 pro-Ukraine authors are AutoNewsAdmin, AutoNewspaperAdmin, rotoreuters, followed by en and autotldr.

Again, for the same author, the number pro-Ukraine posts is about double the number of pro-Russia posts.

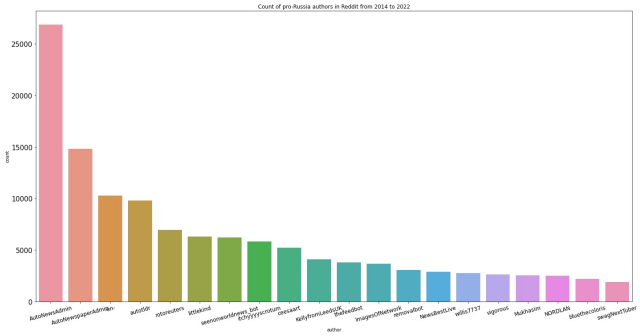

Figure 15: Count of pro-Russia posters

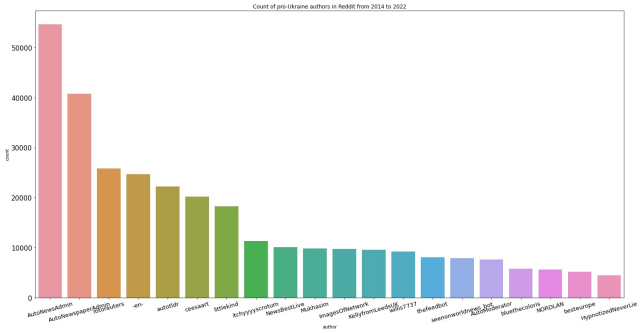

Figure 16: Count of pro-Ukraine posters

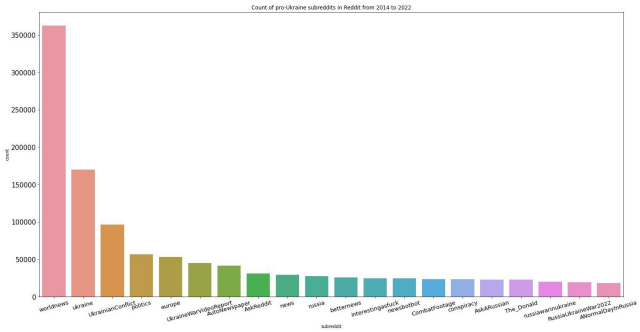

Figure 17: Count of pro-Ukraine subreddits

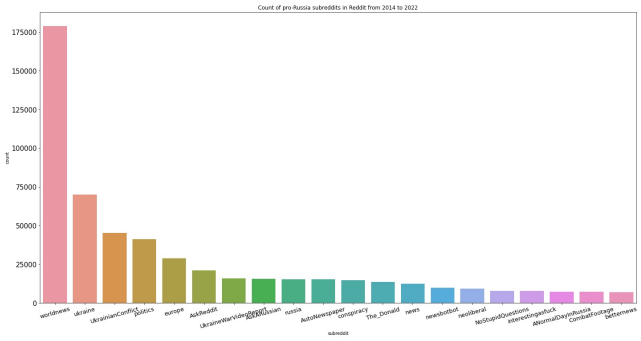

Figure 18: Count of pro-Russia subreddits

4.7 Visualization

4.7.1 *Diagrams*

Diagrams in the project were generated and styled by *matplotlib*[6]. Diagrams are displayed in other sections for illustration purpose.

4.7.2 *Website*

We adopted scrollytelling[13] as the form of our website. When scrolling down the website, the sidebar would indicate what event happened at a certain time. Combined with the charts on the right side, readers have a better understanding of the effect of historical events on Reddit comments and submissions. A brief preview of the website is in Figure 19 and Figure 20.

The technique under the hood is tailwind CSS for styling, React.js for interaction, and React Spring for animations.

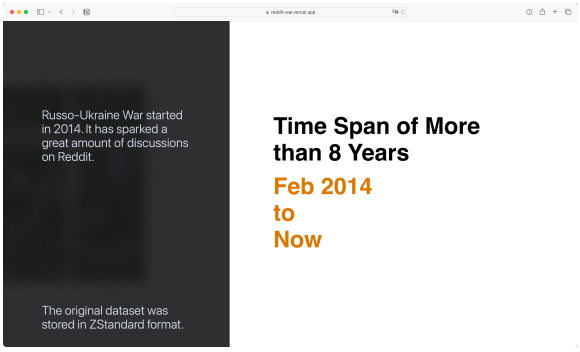

Figure 19: Introduction to the data

Figure 20: Display of events and charts

5. CONCLUSIONS

In this project on reddit, we complete our project expectations. We explore reddit data and keep the data we care about. After comparing Yake and BM25, we choose Yake to filter the topic to get posts about the Ukraine war. We manually label over 3500 data and construct two pipelines to complete sentiment analysis and political standpoint analysis respectively. We count the number and sentiment of submissions and comments at different times. We also count the number of different standpoints and their changes over time and get word clouds. We find the posters and subreddits that generate the most relevant discussions.

6. PERSONAL RESPONSIBILITY

Name	Content
Xingwen	Section 3, 4.2.3, 4.3, 4.7.2
Lixiang Zhang	Section 1, 2.1, 4.2.1, 4.5, 4.6, 4.7.1
Dingran Qi	Section 2.2, 2.3, 4.2.2, 4.4, 5

Table 3: Report

Name	Content
Xingwen	Converting submissions and comments to parquet, sentiment analysis, finding out the replies (comments) of threads (submissions), getting data distribution
Lixiang Zhang	Keyword extraction, sampling all the related posts, making word clouds based on political standpoints, analyzing the number of posters and posts overtime, analyzing the most relevant subreddits and the most prolific authors
Dingran Qi	Keyword extraction for BM25, BM25 testing, sampling submissions and comments, political standpoint analysis, timestamp conversion

Table 4: Codes

Name	Content
Xingwen	Static website
Lixiang Zhang	All the diagrams

Table 5: Visualizations

7. REFERENCES

[1] A. Baltas, A. Kanavos, and A. K. Tsakalidis. An apache spark implementation for sentiment analysis on twitter data. In *International Workshop of Algorithmic Aspects of Cloud Computing*, pages 15–25. Springer, 2016.

[2] R. Campos, V. Mangaravite, A. Pasquali, A. Jorge, C. Nunes, and A. Jatowt. Yake! keyword extraction from single documents using multiple local features. *Information Sciences*, 509:257–289, 2020.

[3] R. Campos, V. Mangaravite, A. Pasquali, A. M. Jorge, C. Nunes, and A. Jatowt. Yake! collection-independent automatic keyword extractor. In G. Pasi, B. Piwowarski, L. Azzopardi, and A. Hanbury, editors, *Advances in Information Retrieval*, pages 806–810, Cham, 2018. Springer International Publishing.

[4] M. Evans, W. McIntosh, J. Lin, and C. Cates. Recounting the courts? applying automated content analysis to enhance empirical legal research. *Journal of Empirical Legal Studies*, 4(4):1007–1039, 2007.

[5] Q. Guo and A. Xiong. Chinese news keyword extraction algorithm based on textrank and word-sentence collaboration. In *International Conference on Computer Engineering and Networks*, pages 556–563. Springer, 2018.

[6] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[7] A. I. Kadhim. Term weighting for feature extraction on twitter: A comparison between bm25 and tf-idf. In *2019 International Conference on Advanced Science and Engineering (ICOASE)*, pages 124–128, 2019.

[8] H. S. Khawaja. Hugginglearners/russia-ukraine-conflict-articles · datasets at hugging face.

[9] V. Kocaman and D. Talby. Spark nlp: Natural language understanding at scale. *Software Impacts*, page 100058, 2021.

[10] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. *arXiv preprint cs/0205070*, 2002.

[11] A. Rao and N. Spasojevic. Actionable and political text classification using word embeddings and lstm. *arXiv preprint arXiv:1607.02501*, 2016.

[12] M. Rathi, A. Malik, D. Varshney, R. Sharma, and S. Mendiratta. Sentiment analysis of tweets using machine learning approach. In *2018 Eleventh international conference on contemporary computing (IC3)*, pages 1–3. IEEE, 2018.

[13] D. Seyser and M. Zeiller. Scrollytelling–an analysis of visual storytelling in online journalism. *22nd international conference information visualisation (IV). IEEE*, pages 401–406, 2018.

[14] P. D. Turney. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. *arXiv preprint cs/0212032*, 2002.

[15] B. Yu, S. Kaufmann, and D. Diermeier. Classifying party affiliation from political speech. *Journal of Information Technology & Politics*, 5(1):33–48, 2008.