Up2Date Research: A personalized recommendation application for researchers

by

Christopher James Bobotsis

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Data Science

Waterloo, Ontario, Canada, 2022
©Christopher James Bobotsis 2022

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In our current age, the number of academic papers published each year is growing at an exponential rate [1]. Staying current in one's respective field can not only be overwhelming, but a struggle to achieve.

To help reach their goals, this paper introduces "Up2Date Research", a platform for academics to stay up to date in their field of research by receiving personalized recommendations using the predictive power of machine learning models. Up2Date Research provides users with six different sources of information including Arxiv, PMLR, Towards-Data-Science, Reddit, Twitter and YouTube.

Acknowledgements

I would like to thank my supervisor Kimon Fountoulakis for supporting me in the completion of this Master's Thesis and my family for being supportive of me taking the initiative to further my education.

Dedication

This paper, as small a token as it is in comparison to him, is dedicated to my late father Eustratios Bobotsis. May we meet again in the Kingdom of Heaven.

Table of Contents

Author's Declaration ii

Abstract iii

Acknowledgements iv

Dedication v

List of Figures xi

List of Tables xv

List of Abbreviations xvi

1 Background 1

1.1 What is the problem? . . . 1

1.2 Why is this important? . . . 1

1.3 Why don't previous methods work on that problem? . . . 1

1.4 What is the solution to the problem the authors propose? . . . 2

1.4.1 Login Page . . . 2

1.4.2 Recommendations Page . . . 5

1.4.3 Likes Page . . . 10

1.4.4 Papers Page . . . 12
1.4.5 Settings Page . . . 16
1.4.6 Keywords Page . . . 19
1.4.7 Sign-Up Page . . . 20
1.5 What interesting research questions does the paper raise? . . . 22
1.6 Datasets . . . 22
1.6.1 User Sample Datasets . . . 22
1.6.2 Proxy Datasets . . . 23
1.7 Infrastructure . . . 23
1.8 Data . . . 23

2 Related Work **24**
2.1 Arxiv Sanity . . . 24
2.2 Google Scholar . . . 25
2.3 Research Rabbit . . . 27
2.4 Deeplearn . . . 28
2.5 Up2Date Research vs. Related Work . . . 28

3 Research Questions **30**
3.1 What Linear Model and Text Vectorizer should be used? . . . 30
3.2 Should per-source or combined-source training be used? . . . 31
3.3 What Metric Performance can be achieved on the Sample Dataset? . . . 31
3.4 What is the Metric Performance difference between the Linear and Transformer Models? . . . 31
3.5 What is the effect on Metric Performance as training size increases? . . . 32
3.6 What is the effect of Classification Threshold on Metric Performance? . . . 32
3.7 Do the False Positive and False Negative predictions make sense? . . . 32
3.8 What is the performance on future recommendations based on past labels? 32

4 Methods **34**
4.1 Experiment Framework . . . 34
4.2 Models . . . 36
4.3 Text Encoders . . . 40
4.4 Experiment Types . . . 46
4.5 Metrics . . . 46

5 Evaluation **48**
5.1 Choosing a Linear Model and Text Vectorizer . . . 48
5.2 Model Performance on the Sample Dataset . . . 54
5.3 Model Performance on Increasing Training Sizes . . . 58
5.4 Classification Threshold Effects on Metric Performance . . . 68
5.5 Analysis of Model Predictions . . . 71
5.6 Temporal Analysis of Sample Dataset Performance . . . 92

6 Conclusion **95**

References **96**

APPENDICES **99**

A Infrastructure **100**
A.1 Client . . . 100
A.1.1 Design . . . 100
A.1.2 Leveraging Local Storage . . . 100
A.2 Backend . . . 103
A.2.1 REST API . . . 103
A.2.2 Sessions . . . 124
A.3 Database . . . 125

A.3.1 Tables . . . 126
A.3.2 Foreign Keys . . . 131
A.3.3 Accessing the Database . . . 133
A.3.4 Database Users . . . 133
A.3.5 Normalized vs. Denormalized Tables . . . 135
A.4 Remote Server . . . 135
A.4.1 Fetching User Data . . . 136

B Data **140**
B.1 Data Collection . . . 141
B.1.1 Twitter . . . 141
B.1.2 Reddit . . . 144
B.1.3 Arxiv . . . 146
B.1.4 PMLR . . . 148
B.1.5 Towards Data Science . . . 148
B.1.6 Youtube . . . 149
B.2 Data Processing . . . 150

C Data Collection and Samples **151**
C.1 Data Collection of Sources . . . 151
C.1.1 Fetching User Data . . . 152
C.1.2 Master Fetch Towards Data Science . . . 153
C.1.3 Master Fetch PMLR . . . 154
C.1.4 Twitter User Data Fetching . . . 155
C.1.5 Reddit User Data Fetching . . . 156
C.1.6 PMLR User Data Fetching . . . 157
C.1.7 Arxiv User Data Fetching . . . 158
C.1.8 Youtube User Data Fetching . . . 159

C.1.9 Towards Data Science Data Fetching . . . 160
C.2 Source Data Samples . . . 161
C.2.1 Twitter . . . 161
C.2.2 Reddit . . . 168
C.2.3 PMLR . . . 173
C.2.4 Arxiv . . . 182
C.2.5 Youtube . . . 186
C.2.6 Towards Dada Science . . . 187

D Future Work **192**
D.1 Modeling . . . 192
D.1.1 Fixing False Positives and False Negatives . . . 192
D.2 Client . . . 192
D.2.1 Streamline New User Integration . . . 192
D.3 Analytics . . . 193
D.3.1 Model Performance Monitoring . . . 193
D.3.2 Data Download Monitoring . . . 193
D.4 Database . . . 193
D.4.1 Table Denormalization . . . 193

Glossary **194**

List of Figures

1.1 Logged-Out NavBar 2
1.2 Logged-In NavBar 2
1.3 Login Page 3
1.4 Login Page Error Message 4
1.5 Recommendations Page 6
1.6 Recommendations Page Date Filtering 7
1.7 Recommendations Page Keywords Filtering 8
1.8 Arxiv Post 9
1.9 Voted Arxiv Post 10
1.10 Likes Page - Disliking a Previously Liked Post 11
1.11 Papers Page - Submitting URLs 13
1.12 Papers Page - Invalid URLs 14
1.13 Papers Page - Remove Invalid URL (red box) and Saving (blue box) 15
1.14 Papers Page - List of Valid Papers 16
1.15 Settings Page 17
1.16 Settings Page for Twitter 18
1.17 Settings Page Arxiv 19
1.18 Keywords Page 20
1.19 Sign-Up Page 21

2.1 Arxiv Sanity Page with a tag of "diffusion models" (cyan box). Labels can be chosen with + or - (magenta box). 25
2.2 Enabling Email Alerts for Google Scholar 26
2.3 The user interface for research rabbit. Outlined is the collections of papers (magenta box), selected collection (cyan box), and work related to the selected collection (orange box). 27
2.4 User Interface for deeplearn.org 28
2.5 Venn Diagram of Related Work and Up2Date Research - Arxiv Sanity (A), Google Scholar (B), Research Rabbit (C), Deeplearn (D) and Up2Date Research (E) 29

5.1 Linear Model and Encoder Combination Results Averages 49
5.2 TFIDF Encoder Times 50
5.3 Universal Sentence Encoder Times 51
5.4 Universal Sentence Encoder Times Without Start 52
5.5 Combined Source Predictions for Linear Model 55
5.6 Combined Source Predictions for Transformer Model 56
5.7 Combined Source Predictions Comparisons 57
5.8 Increasing Train Size Transformer and Linear - 2000 data points 59
5.9 Increasing Train Size Transformer and Linear Difference - 2000 data points 61
5.10 Increasing Train Size Transformer and Linear - All data points 63
5.11 Increasing Train Size Transformer and Linear Difference - All data points . 64
5.12 Increasing Train Size on Proxy Dataset - Transformer Model 66
5.13 Increasing Train Size on Proxy Dataset - Linear Model 67
5.14 Metric vs. Classification Threshold - Linear Model 69
5.15 Metric vs. Classification Threshold - Transformer Model 70
5.16 Linear Model False Positives 1 to 5 72
5.17 Linear Model False Positives 6 to 10 73
5.18 Linear Model False Positives 11 to 13 74

5.19 Linear Model False Positives 14 to 16 75
5.20 Linear Model False Positives 17 to 18 76
5.21 Linear Model False Positive 19 76
5.22 Linear Model False Negatives 1 to 4 79
5.23 Linear Model False Negatives 5 to 8 80
5.24 Transformer Model False Positives 1 to 4 82
5.25 Transformer Model False Positives 5 to 6 83
5.26 Transformer Model False Positives 7 to 8 84
5.27 Transformer Model False Positives 9 to 10 85
5.28 Transformer Model False Positive 11 86
5.29 Transformer Model False Negatives 1 to 4 88
5.30 Transformer Model False Negatives 5 to 8 89
5.31 Transformer Model False Negatives 9 90
5.32 Transformer Model False Negatives 10 90
5.33 Transformer Model False Negative 11 91
5.34 Temporal and Non-Temporal Test F1 Scores - Linear Model 93
5.35 Temporal and Non-Temporal Test F1 Scores - Transformer Model 94

A.1 Unlabelled Post 111
A.2 Labelled Post 112
A.3 Papers Page - Labeling a Paper 121
A.4 Database Schema 132

B.1 Sample Tweet 142
B.2 Sample Tweet with Reply 143

C.1 User Data Fetch Flowchart 152
C.2 Towards Data Science - Master Data Fetch Flowchart 153
C.3 PMLR - Master Data Fetch Flowchart 154

C.4 Twitter Data Fetch Flowchart 155
C.5 Reddit Data Fetch Flowchart 156
C.6 PMLR Data Fetch Flowchart 157
C.7 Arxiv Data Fetch Flowchart 158
C.8 Youtube Data Fetch Flowchart 159
C.9 Towards Data Science Data Fetch Flowchart 160

List of Tables

4.1 Experiment Framework Arguments 36

5.1 Analysis of Linear Model False Positives 78
5.2 Analysis of Linear Model False Negatives 81
5.3 Analysis of Transformer Model False Positives 87
5.4 Analysis of Transformer Model False Negatives 92

A.1 Cronjobs for Data Collection and Model Training 139

B.1 Source Details 140

List of Abbreviations

API Application Programming Interface 140

AWS Amazon Web Services 135

EC2 Elastic Cloud Compute 135

PMLR Proceedings of Machine Learning iv, 2, 136, 140, 148

URI Uniform Resource Identifier 103–105, 107–110, 114, 115, 117, 119, 120, 123

Chapter 1

Background

1.1 What is the problem?

Due to the number of academic papers published each year growing at an exponential rate [1], staying current in one's respective field can not only be overwhelming, but a struggle to achieve.

1.2 Why is this important?

Academic researchers already have busy lifestyles having to balance teaching, research and mentoring students. If a tool could help reduce their workload, then it should be developed.

1.3 Why don't previous methods work on that problem?

There are a number of current works that aim to address part of this problem including arxiv sanity, google scholar, research rabbit and deeplearn. However, none of these works are able to provide all three key features of providing recommendations, showing multiple sources and allowing for users to label the recommendations they receive. Each of these works, and their shortcomings, will be discussed in detail in Chapter 2.

1.4 What is the solution to the problem the authors propose?

This paper proposes "Up2Date Research", a platform for academics to stay up to date in their field of research by receiving personalized recommendations using the predictive power of machine learning models, to solve this problem. Up2Date Research provides users with six different sources of information including Arxiv, PMLR, Towards-Data-Science, Reddit, Twitter and YouTube. Up2Date Research contains a total of seven pages that are visible to the user. When logged out, they include the sign-up and login pages (Figure 1.1). When logged in, they include the recommendations, likes, papers, settings, keywords and logout pages (Figure 1.2).



Figure 1.1: Logged-Out NavBar

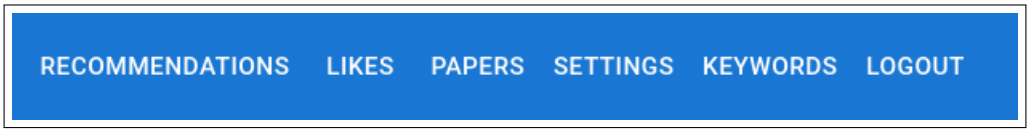


Figure 1.2: Logged-In NavBar

1.4.1 Login Page

When first navigating to the application's url, you will be greeted by a navigation bar that contains the login and sign-up buttons (Figure 1.3).

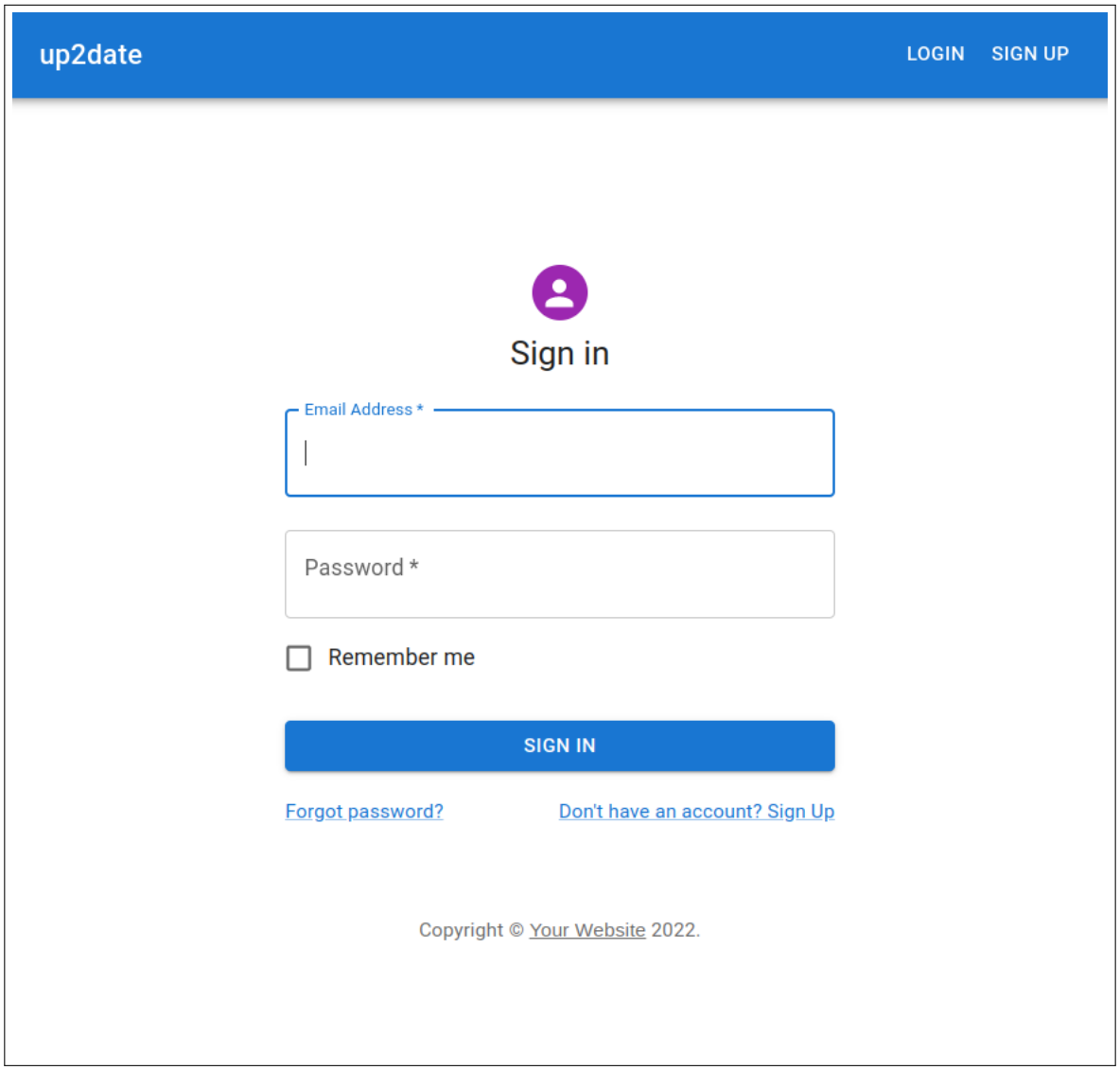


Figure 1.3: Login Page

If either the username or password is incorrect, then the user will be greeted with a warning message (Figure 1.4).

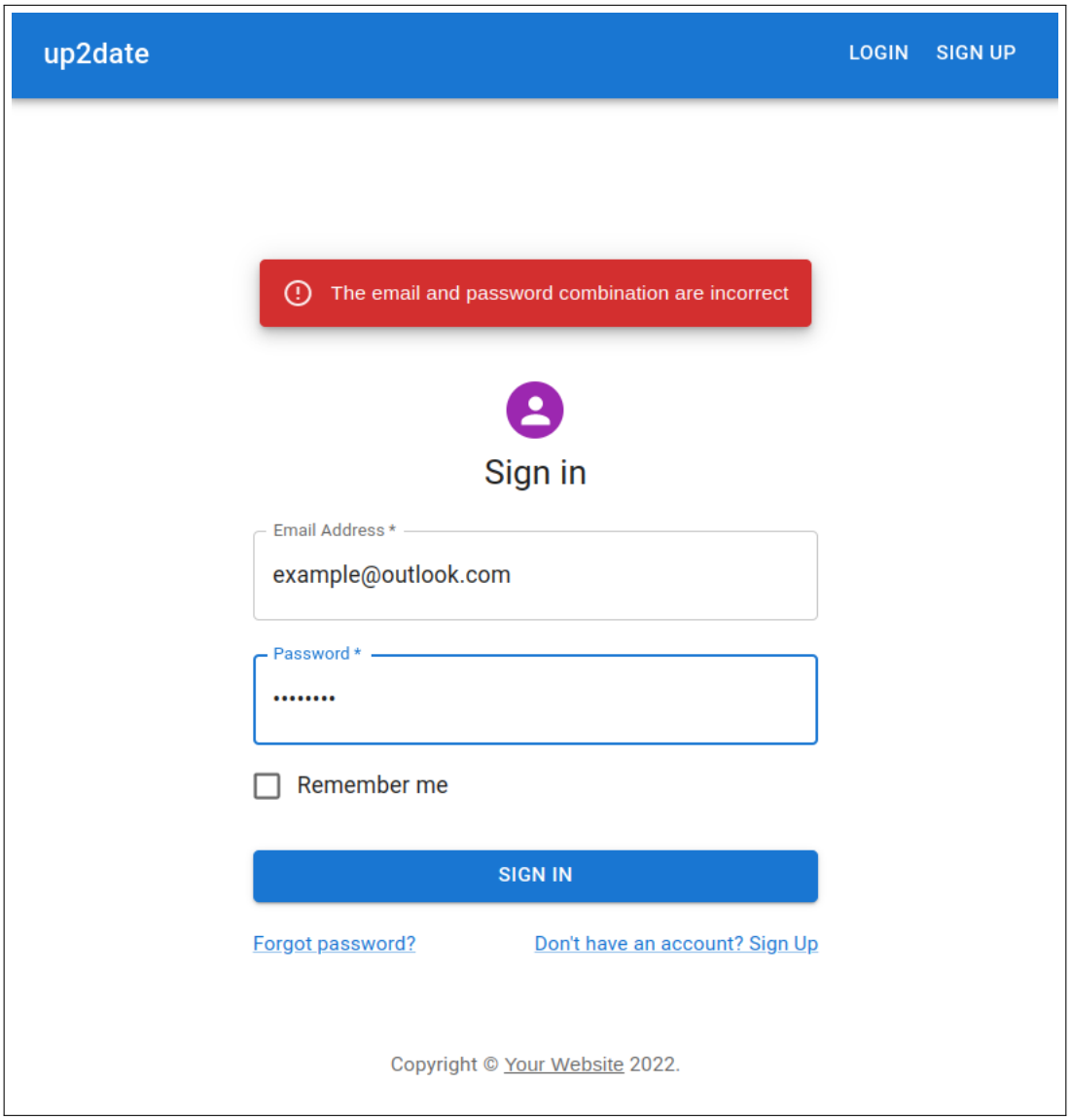


Figure 1.4: Login Page Error Message

Once logged in successfully, the navigation bar will give access to recommendations, likes, papers, settings, keywords and a logout button.

1.4.2 Recommendations Page

The recommendations page is the main attraction of this application. It shows the user all relevant posts from each source in a horizontal scroll window (Figure 1.5).

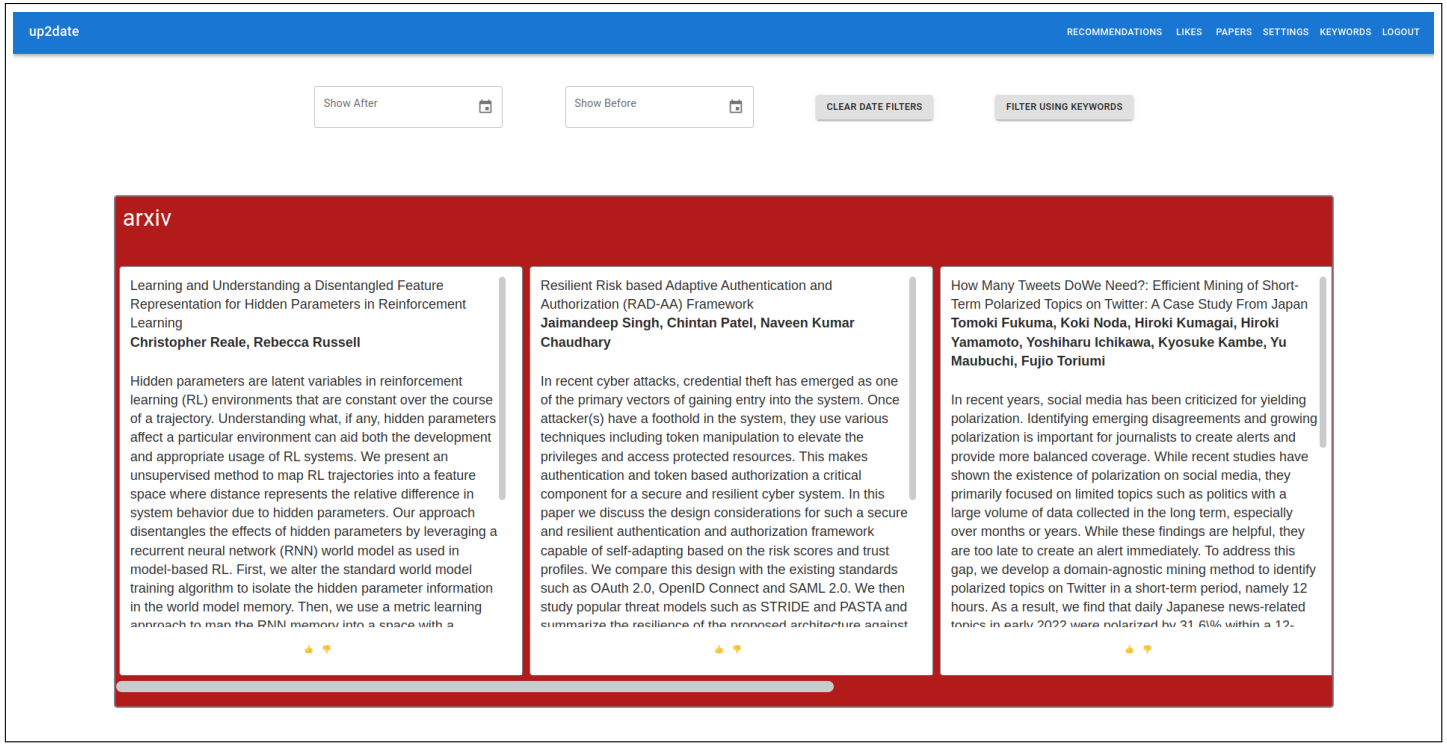


Figure 1.5: Recommendations Page

If the user desires, they are able to filter this data by dates (Figure 1.6) or by their currently selected keywords (Figure 1.7) using the icons at the top of the page.

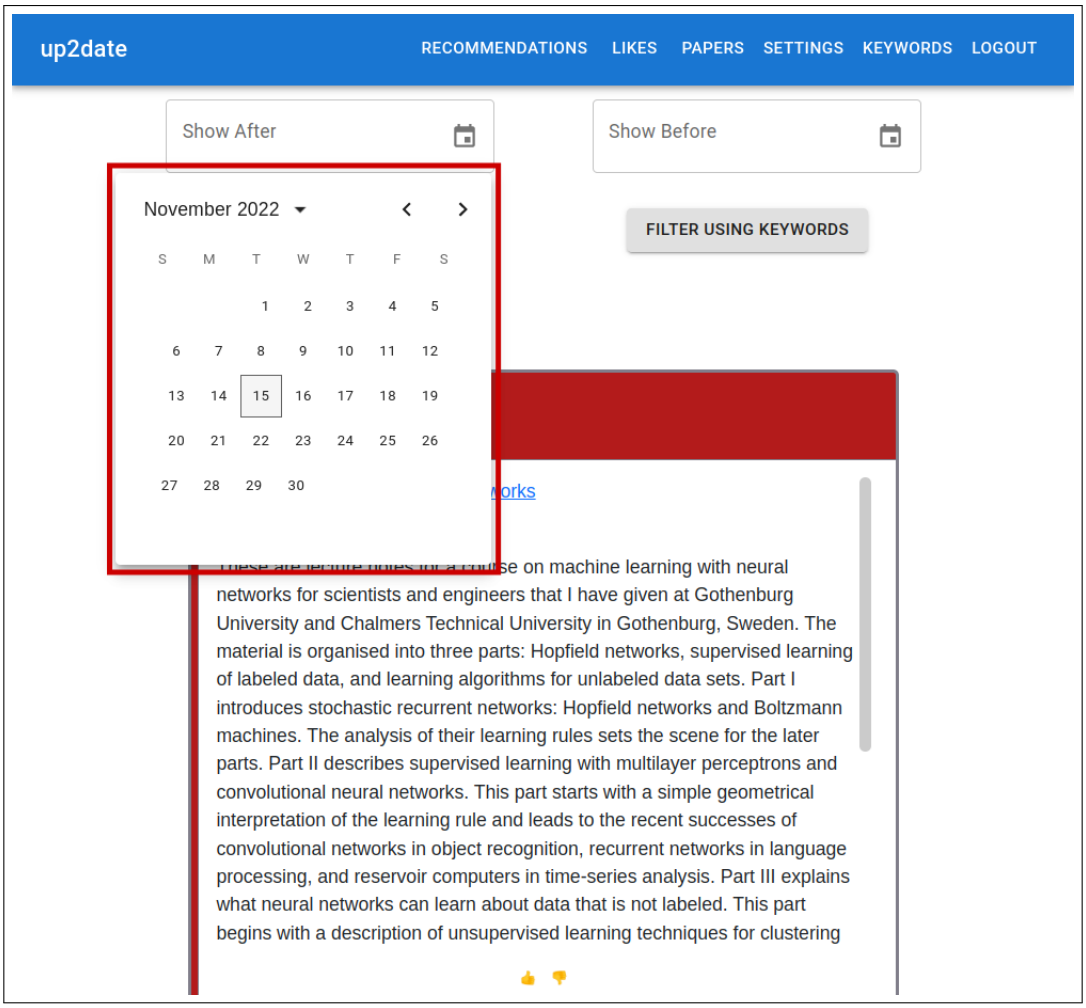


Figure 1.6: Recommendations Page Date Filtering

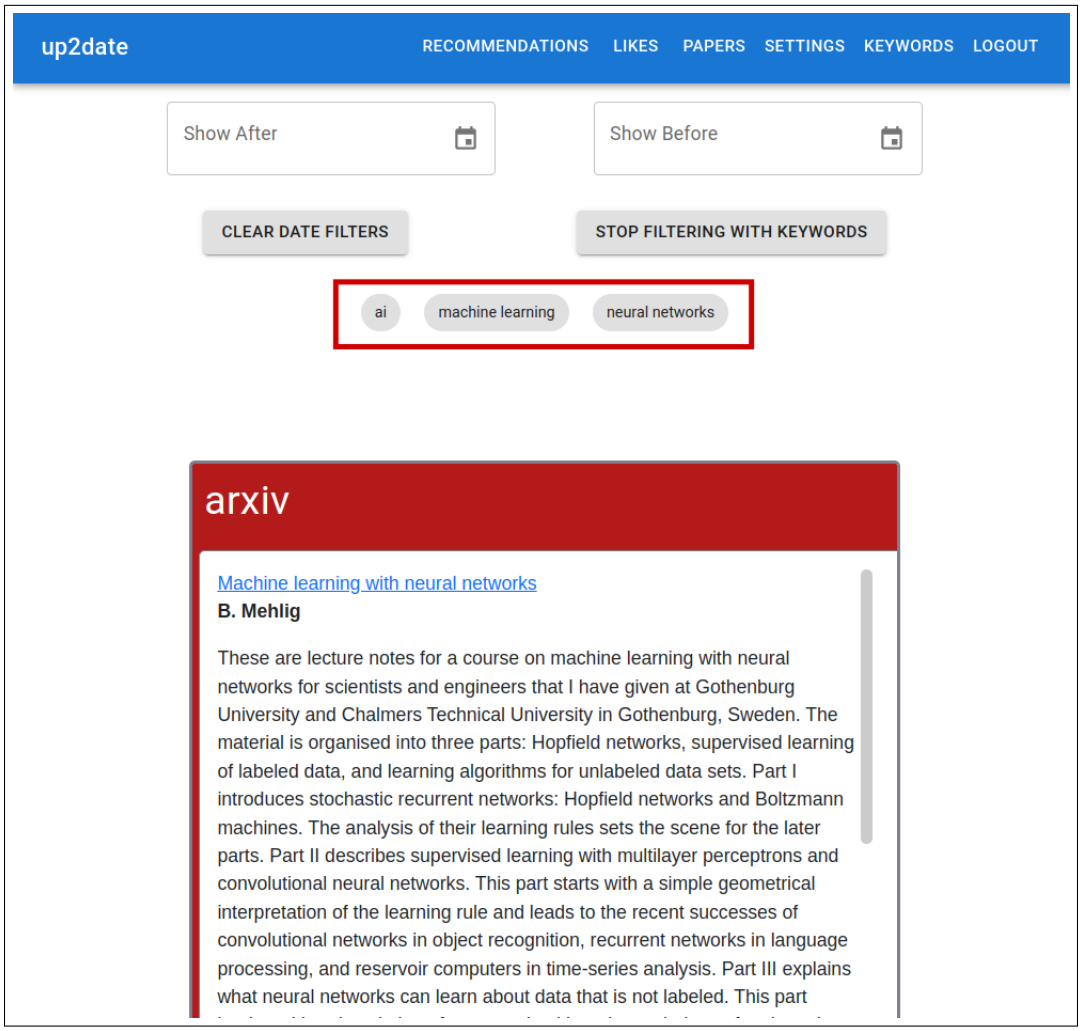


Figure 1.7: Recommendations Page Keywords Filtering

The user is also able to label each post with a thumbs-up, indicating a post they would like to see more of, or thumbs-down, a post they do not want to see more of (Figure 1.8). These labels will be updated on the database.

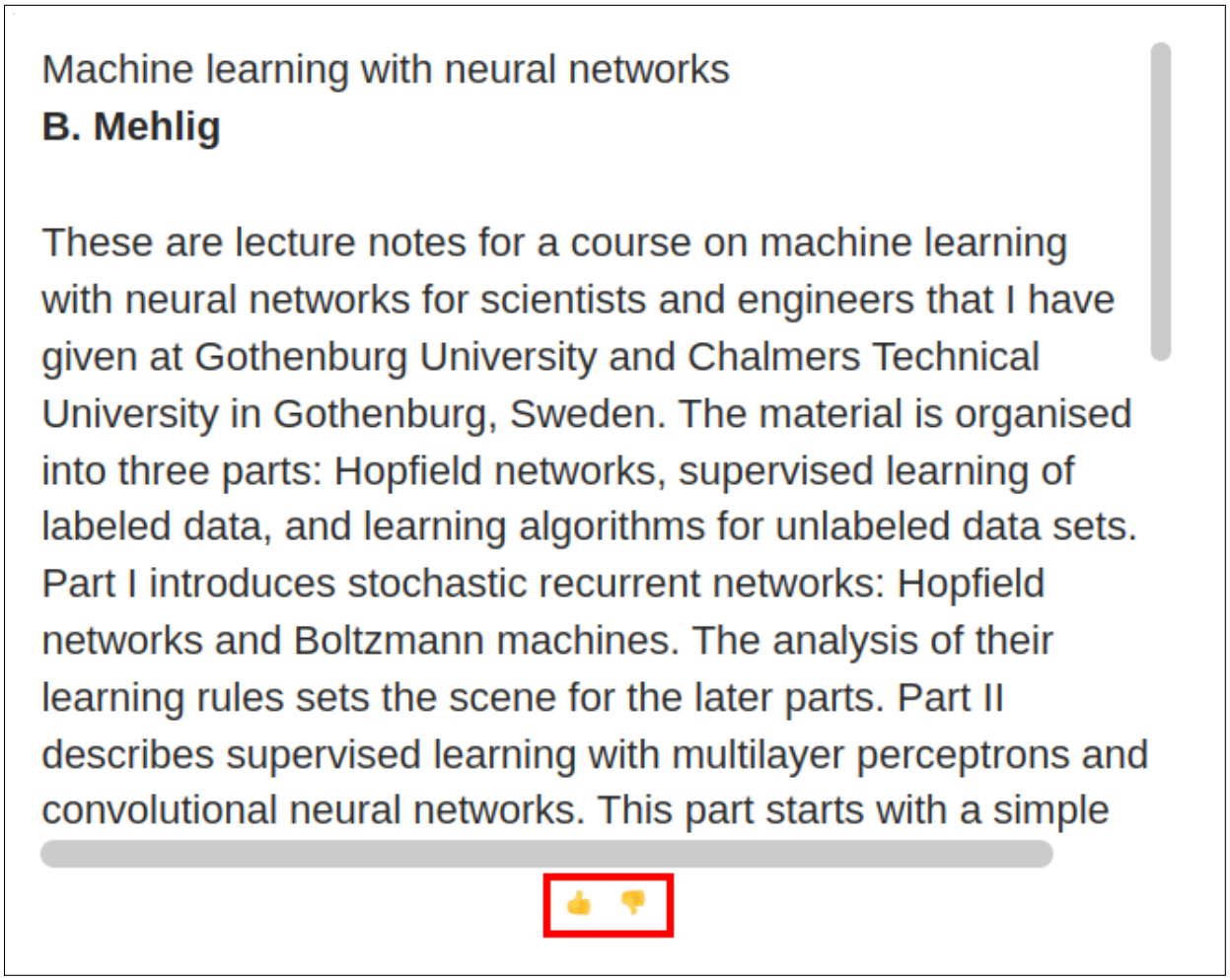


Figure 1.8: Arxiv Post

Once a post has been voted on, it is possible to reverse the vote as well (Figure 1.9); however, if the user changes pages or refreshes the page the post will no longer be visible to reverse the vote.

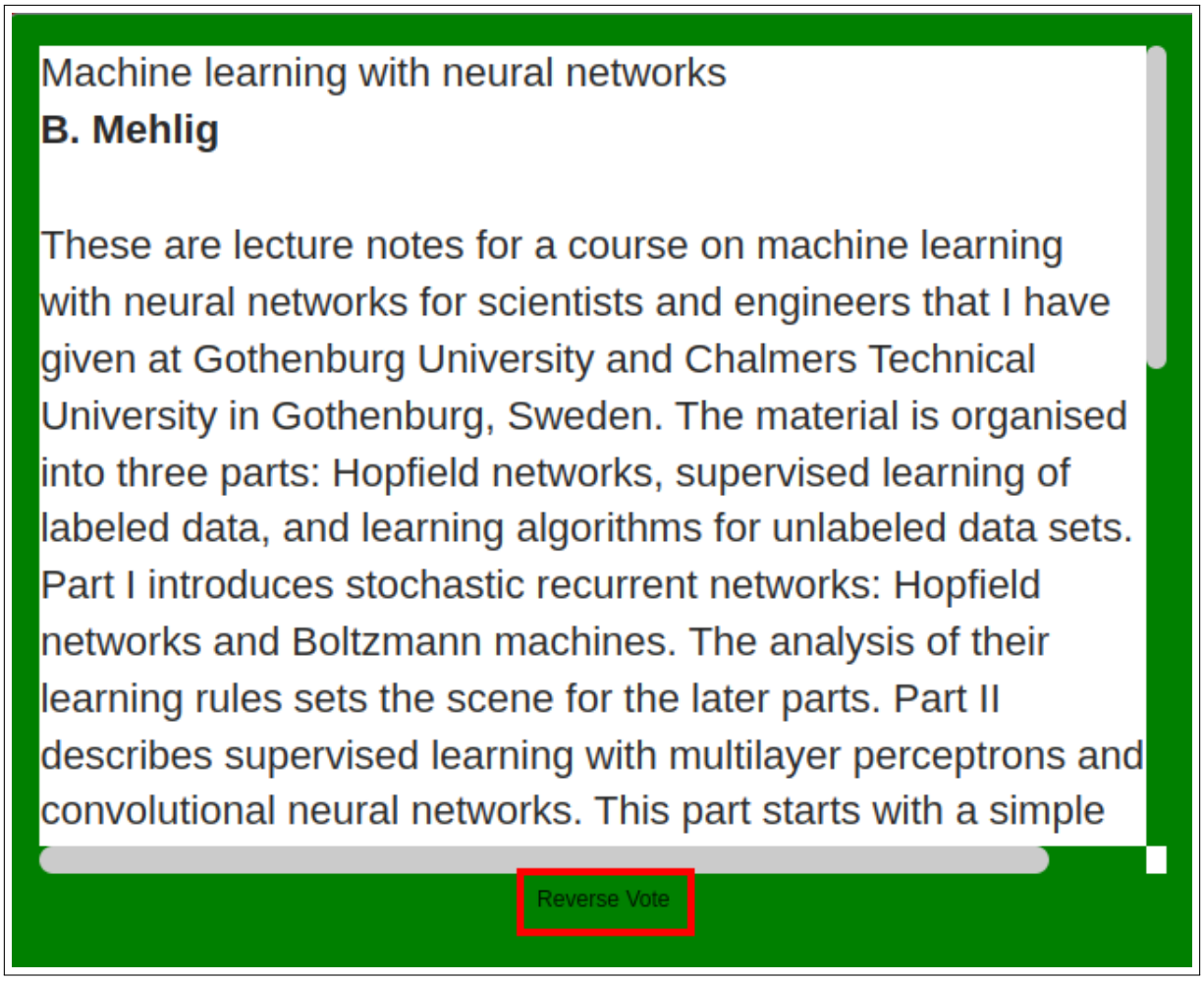


Figure 1.9: Voted Arxiv Post

In order to reduce the number of posts required to be rendered at a time, the application uses pagination. When the user scrolls all the way to the end of the horizontal scroll window for a given source, if there are more posts to show, the backend will send the new data to the client and will become visible to the user.

1.4.3 Likes Page

The likes page is identical to the recommendations page except that it shows all posts, for each source, that a user has already "liked" on the recommendations page. Each of these

posts can now be "disliked", with a thumbs-down, in case the user no longer wants to be recommended similar posts in the future (Figure 1.10).

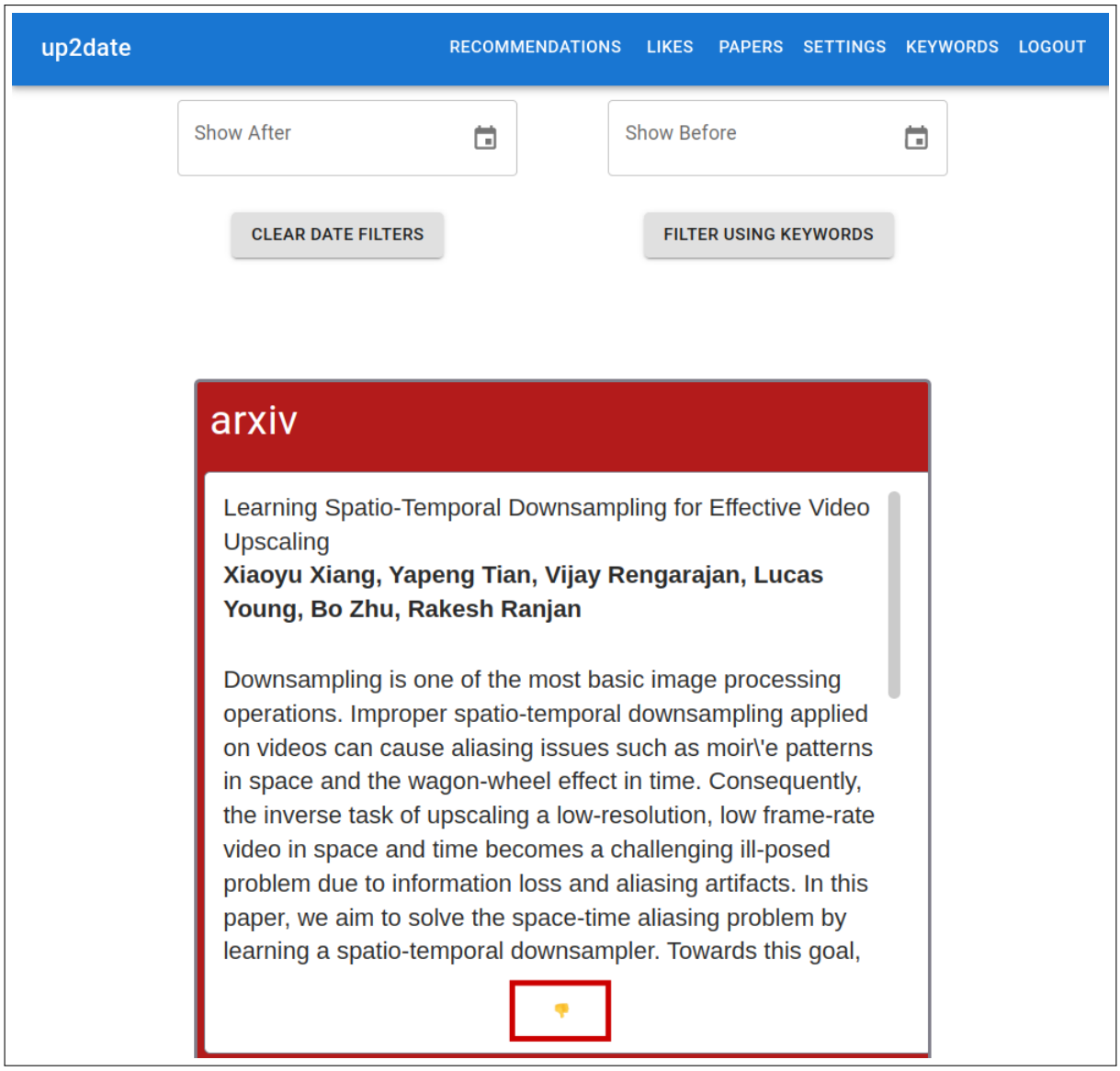


Figure 1.10: Likes Page - Disliking a Previously Liked Post

1.4.4 Papers Page

The papers page allows user's to add a newline delimited list of arxiv.org urls to provide the machine learning model with a larger amount of positively labeled data-points to train on. The user can add the urls for the papers in the text box shown in Figure 1.11. After submitting a list of urls, any invalid urls are shown in the invalid urls section as shown in Figure 1.12. After the user inspects each of these invalid urls, they can remove them from the list and then save the results (Figure 1.13).

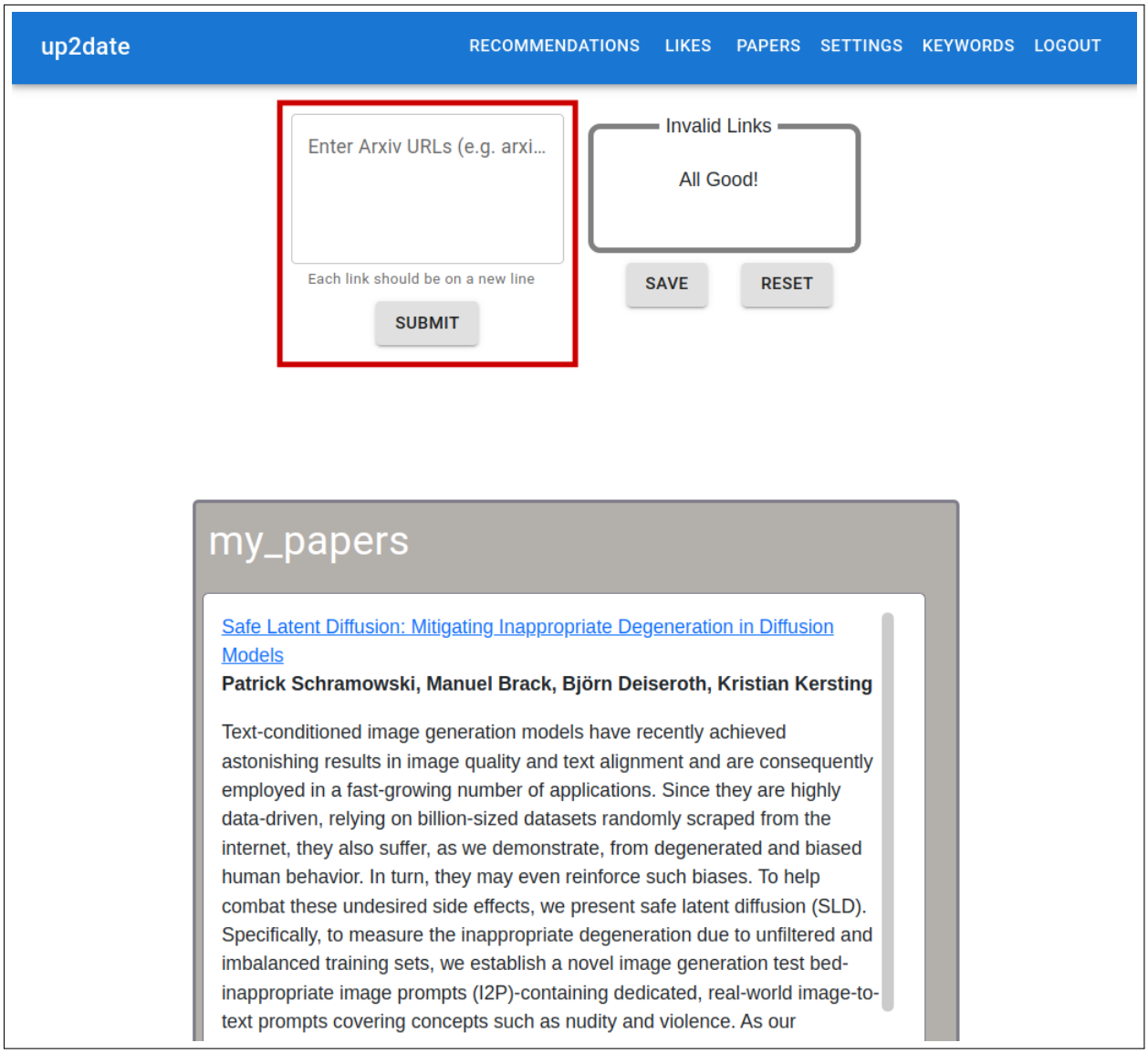


Figure 1.11: Papers Page - Submitting URLs

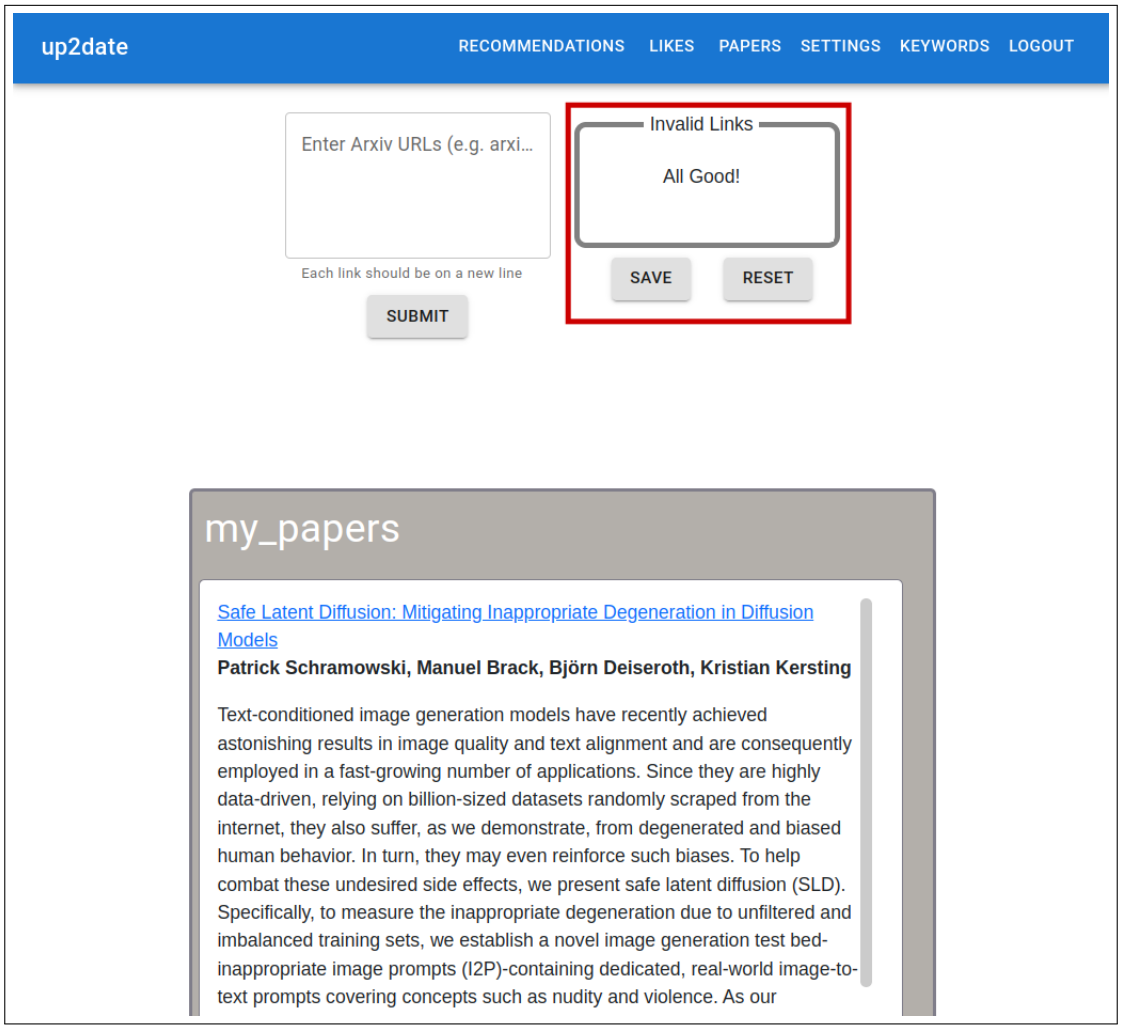


Figure 1.12: Papers Page - Invalid URLs



Figure 1.13: Papers Page - Remove Invalid URL (red box) and Saving (blue box)

Following this, all valid papers will be added to a list as shown in Figure 1.14

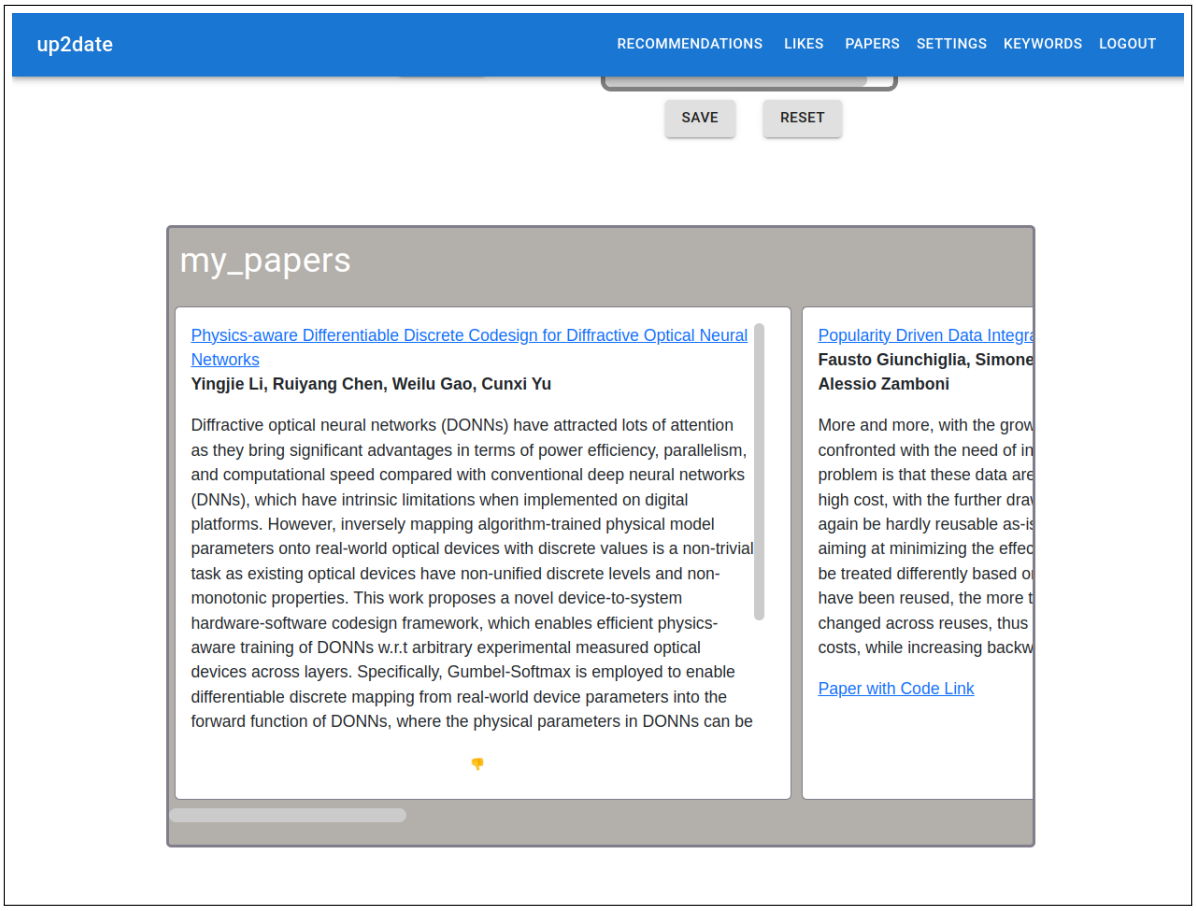


Figure 1.14: Papers Page - List of Valid Papers

Similar to the likes page, the user can press the dislike button on each paper to indicate that they no longer want to be recommended this type of information.

1.4.5 Settings Page

The settings page allows the user to enable or disable the display (on the recommendations and likes pages) and data collection of posts for each source (Figure 1.15). There are two slight variations with both twitter and arxiv.

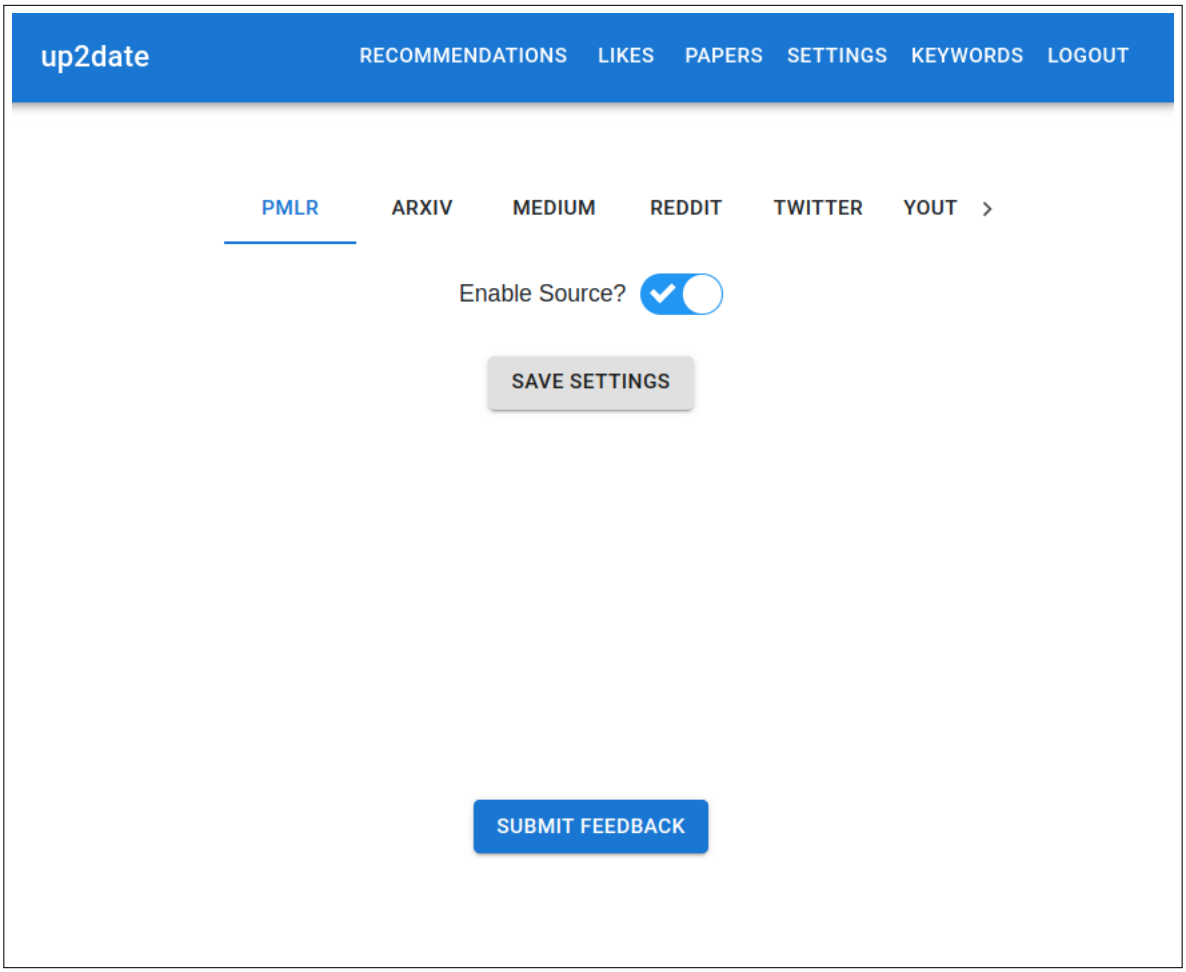


Figure 1.15: Settings Page

Twitter requires the provision of a screen name (Figure 1.16) in order to search for tweets of all the user's followees (who the user follows on twitter).

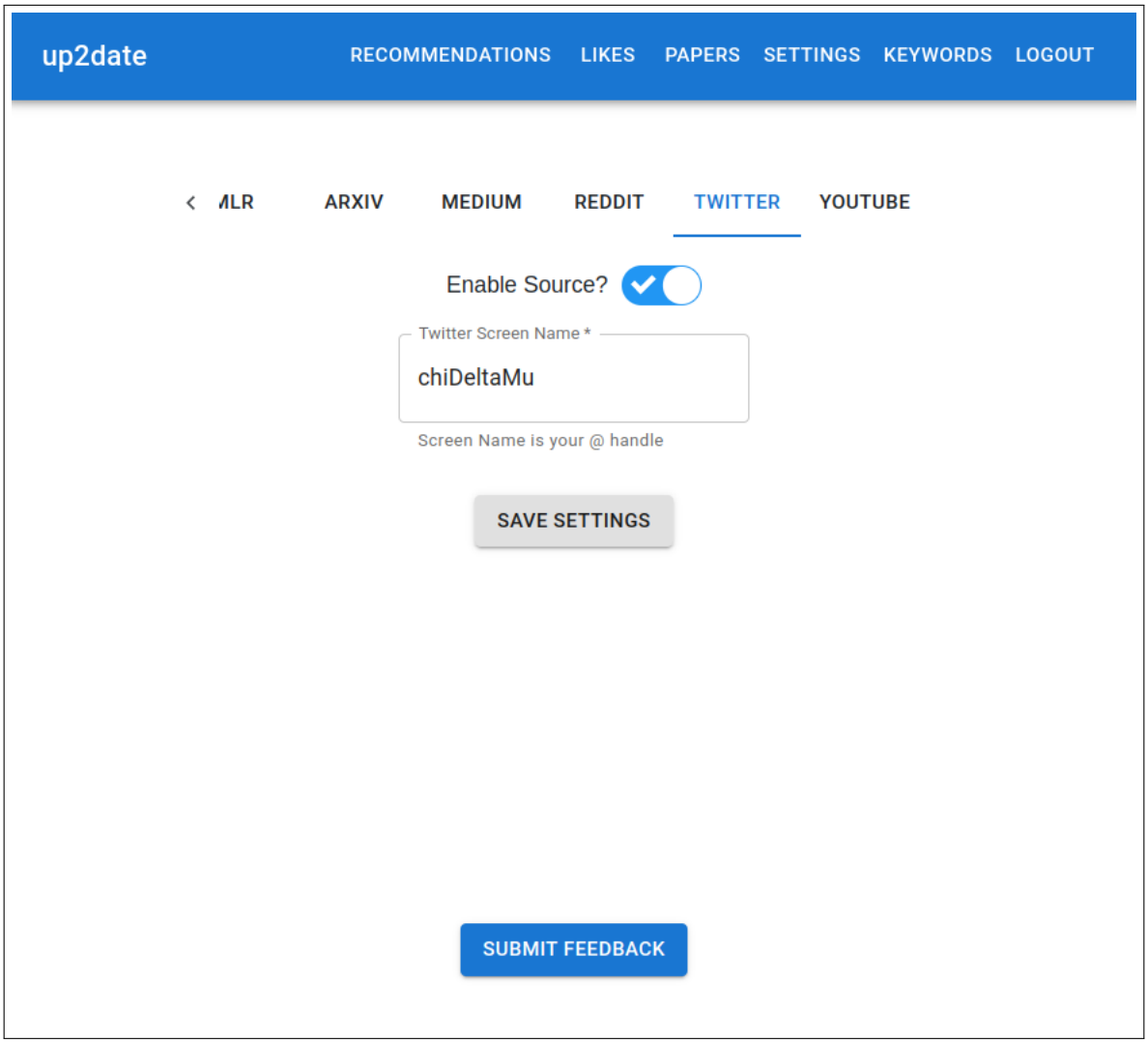


Figure 1.16: Settings Page for Twitter

Arxiv has the addition of being able to choose which sub-categories of fields the user would like to include in the data collection (Figure 1.17). For example, one could only choose to collect data from the Computer Science (CS) field.

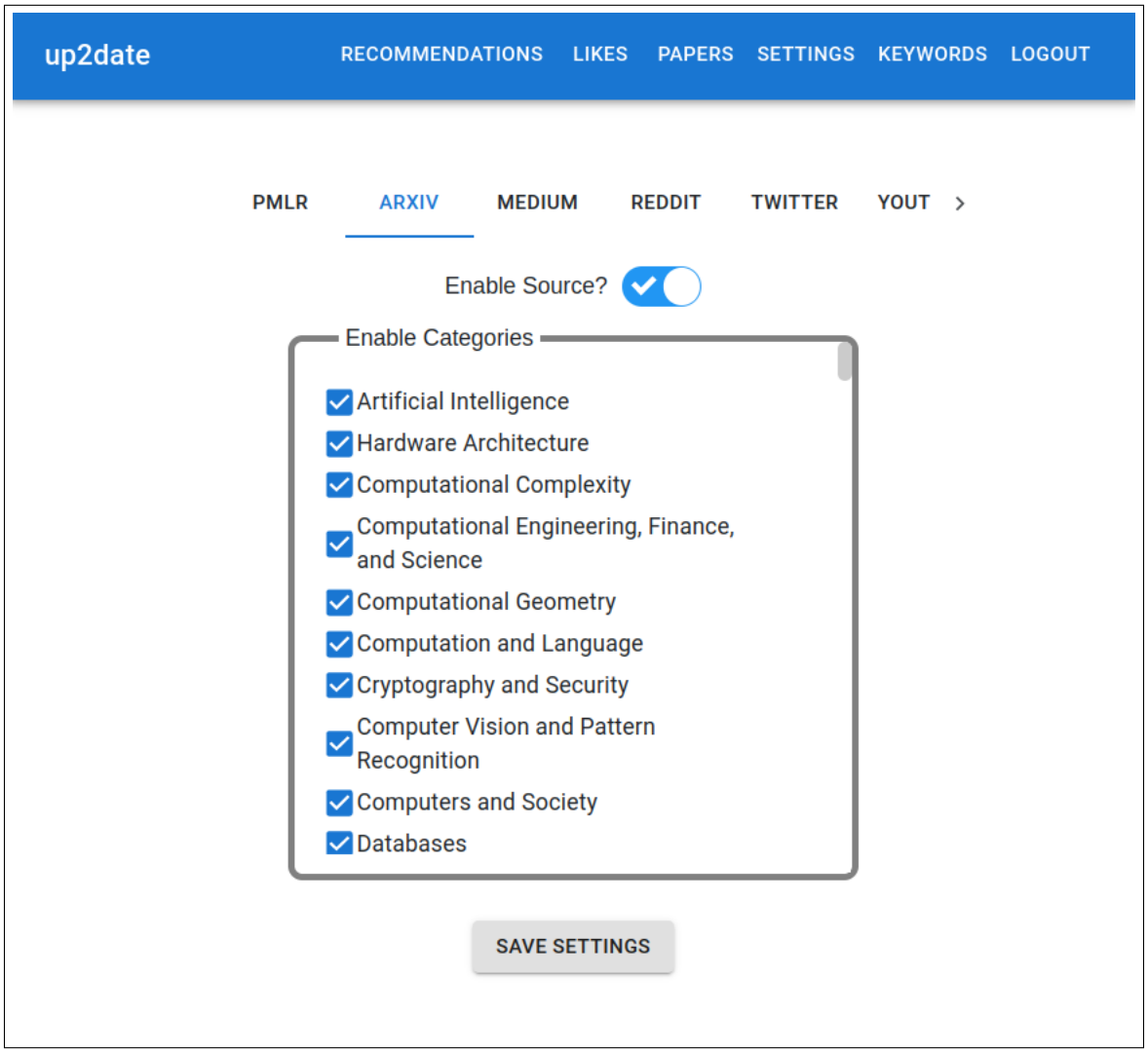


Figure 1.17: Settings Page Arxiv

1.4.6 Keywords Page

The keywords page (Figure 1.18) allows the user to add and delete keywords that they would like to be used in the data collection for each source. To add a keyword, the user types in the input box and then clicks the "ADD" button. To remove a keyword, the user can press the "X" on the keyword button. To persist the changes on the server, the user

can then press “SAVE”.

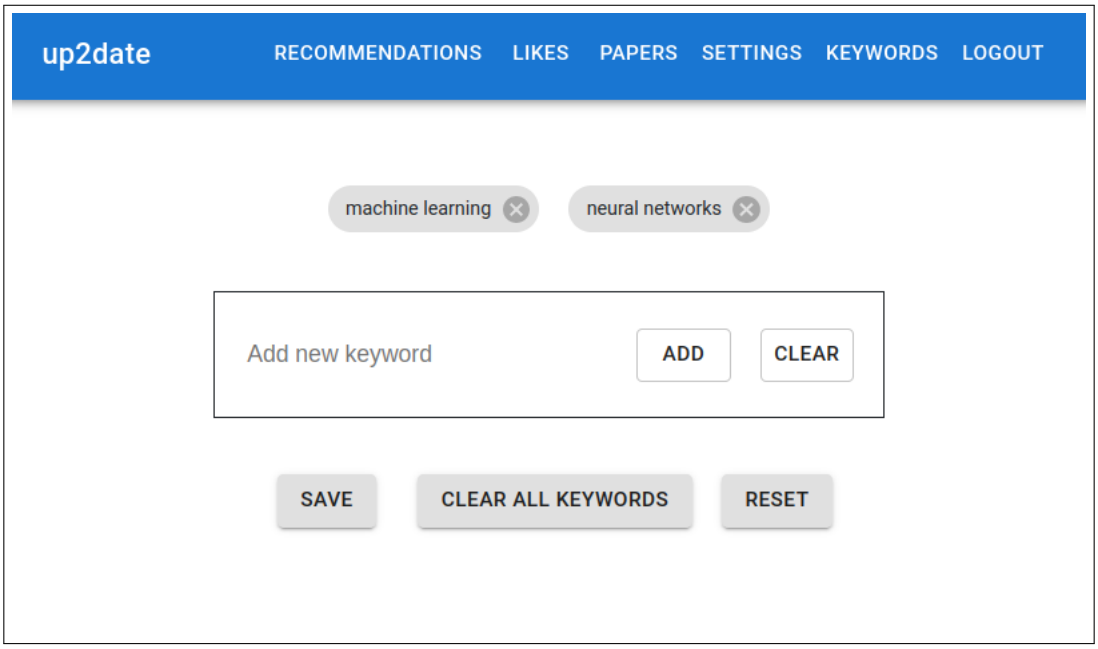


Figure 1.18: Keywords Page

1.4.7 Sign-Up Page

The sign-up page (Figure 1.19) allows users to enter an email address and password that will be added to the database for future sign-ins.

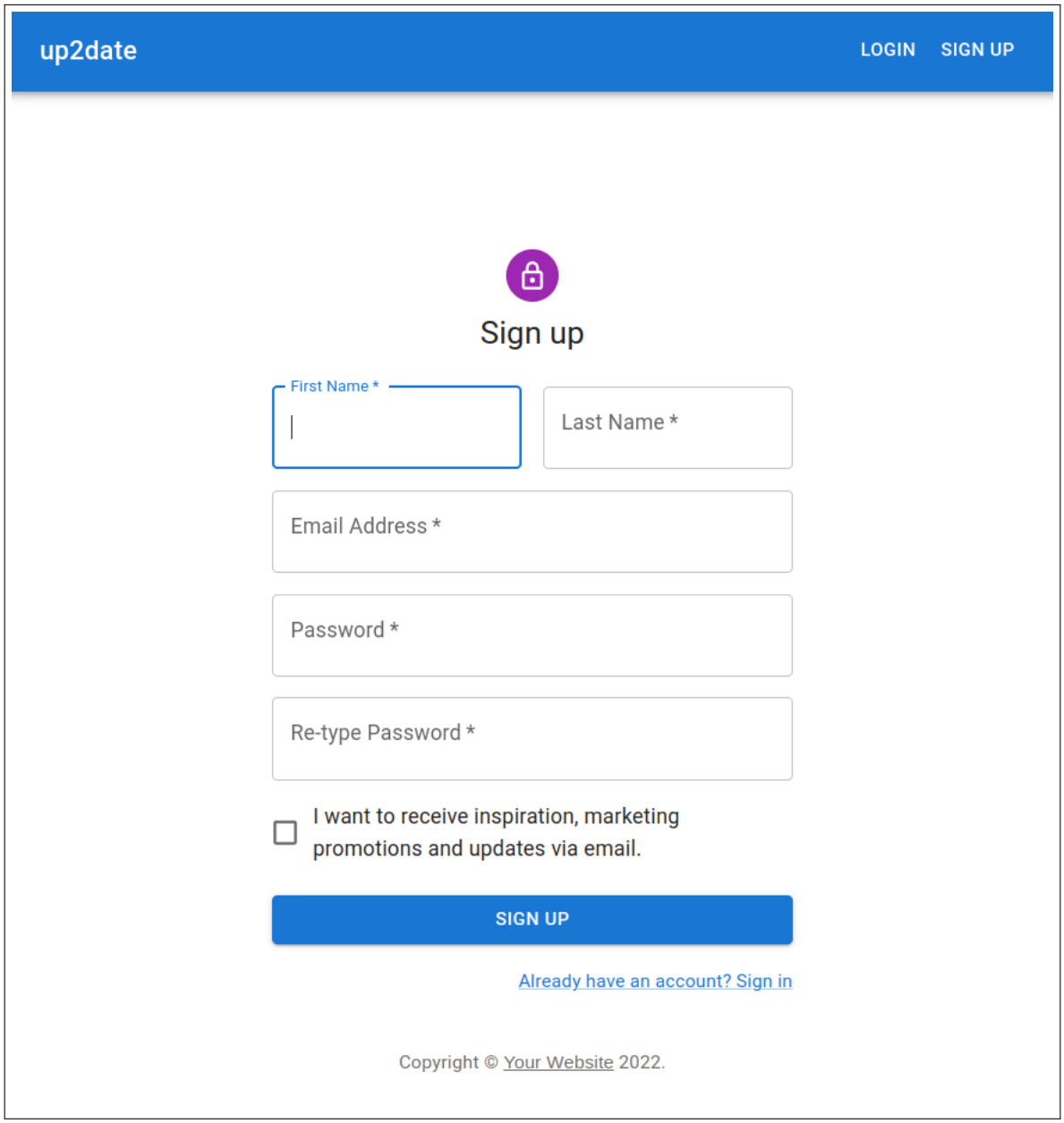


Figure 1.19: Sign-Up Page

1.5 What interesting research questions does the paper raise?

A number of research questions are raised in Chapter 3. They include the following:

1. What Linear Model and Text Vectorizer should be used?
2. Should per source or combined-source training be used?
3. What is the best Metric Performance that can be achieved on the Sample dataset?
4. What is the Metric Performance difference between the Linear and Transformer Models?
5. What is the effect on Metric Performance as training size increases?
6. What is the effect of Classification Threshold on Metric Performance?
7. Do the False Positive and False Negative predictions appear logical?
8. What is the performance on future recommendations based on past labels?

The experiments performed to answer each of these research questions are detailed in Chapter 5.

1.6 Datasets

The data that is used for modeling in this application is textual. There were two types of datasets that were used for this project, user sample datasets and proxy datasets.

1.6.1 User Sample Datasets

User sample datasets were labeled by Professor Kimon Fountoulakis for posts from twitter, reddit, arxiv and pmlr based on what he felt aligned with his research interests. These sample datasets are how an actual user would label posts from the different sources.

1.6.2 Proxy Datasets

Proxy datasets were sourced online by websites such as Kaggle [2] [3]. Proxy datasets are referred to as such as the modeling task is the same, but the distribution of the data and what the labels refer to is not. For example, the user sample dataset's distribution is concentrated within the field of data science and graph learning with the difference in classes being the user's specific research interests. In contrast, the proxy dataset's distribution is much broader and the class differences are based on sentiment (in the case of twitter and reddit) and category (for arxiv) which is likely much easier to separate the two classes. Thus any model performance on the proxy dataset is likely to be an upper bound in comparison to the sample dataset.

Now, the question may be why a proxy dataset is even used if it is only an upper bound. The reason is simple, we are currently limited by the amount of user labeled data to understand how model performance will change as a function of the dataset size. Thus, the proxy dataset is used as the number of data points is an order of magnitude larger than the sample dataset. This will provide an intuition into how model performance will change as more user labels are recorded.

1.7 Infrastructure

The application was written using React and Flask for the client and backend respectively. A MySQL server was used to store all data and Ubuntu sever was used to perform various tasks such as fetching user data. The details of the application's infrastructure can be found in Appendix A.

1.8 Data

Collection of data from the six sources mentioned previously all have different methods to obtain. Details of these processes can be found in Appendix B.

Chapter 2

Related Work

This section aims to review the related work that has a similar goal of Up2Date Research. The features and shortcomings of each work will be described.

2.1 Arxiv Sanity

Arxiv Sanity [4] is a tool that periodically pools the arxiv API for new papers and allows users to tag papers of interest in a web application (Figure 2.1). Afterward, it recommends new papers for each tag using a model. It also allows users to search, rank and sort the recommendations.

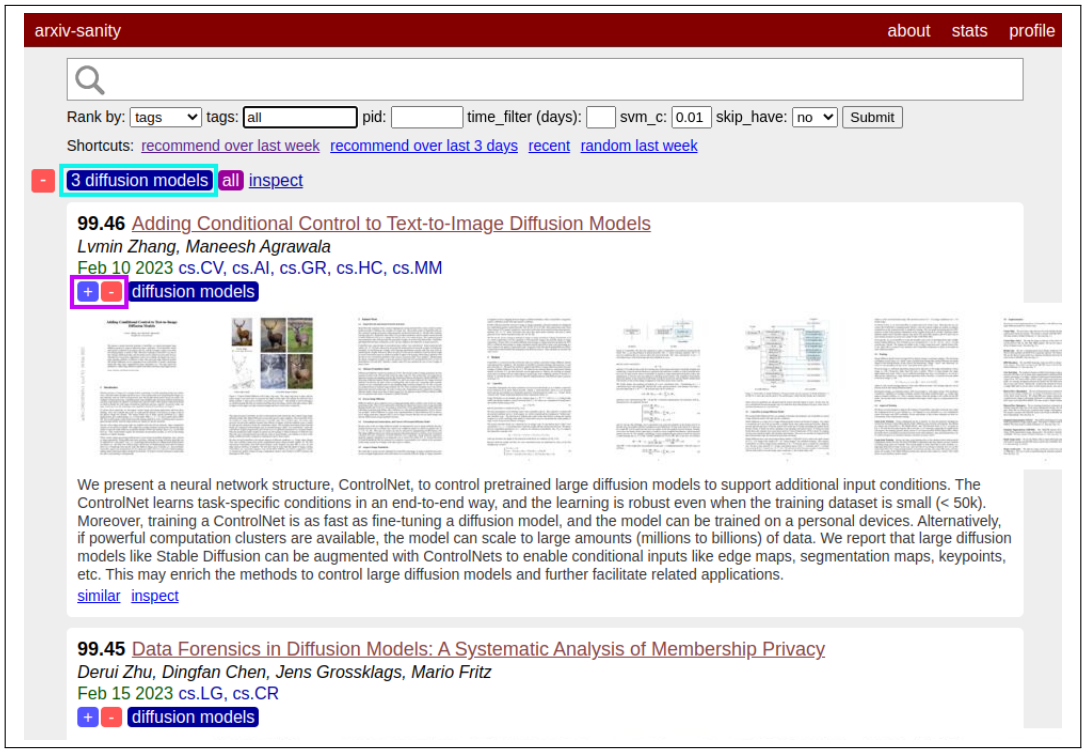


Figure 2.1: Arxiv Sanity Page with a tag of "diffusion models" (cyan box). Labels can be chosen with + or - (magenta box).

Shortcomings

While Arxiv Sanity does provide a method to label papers and provide recommendations using a machine learning model, it is limited to only papers from arxiv. This means it lacks other sources such as reddit and twitter.

2.2 Google Scholar

Google Scholar [5] provides a method to search across many sources such as articles, theses, books, abstracts and court opinions, from academic publishers, professional societies, online repositories, universities and other web sites.

Google Scholar allows the user to creating reading lists and add labels to lists as well. Additionally, email alerts for given keywords can be enabled as shown in Figure 2.2.

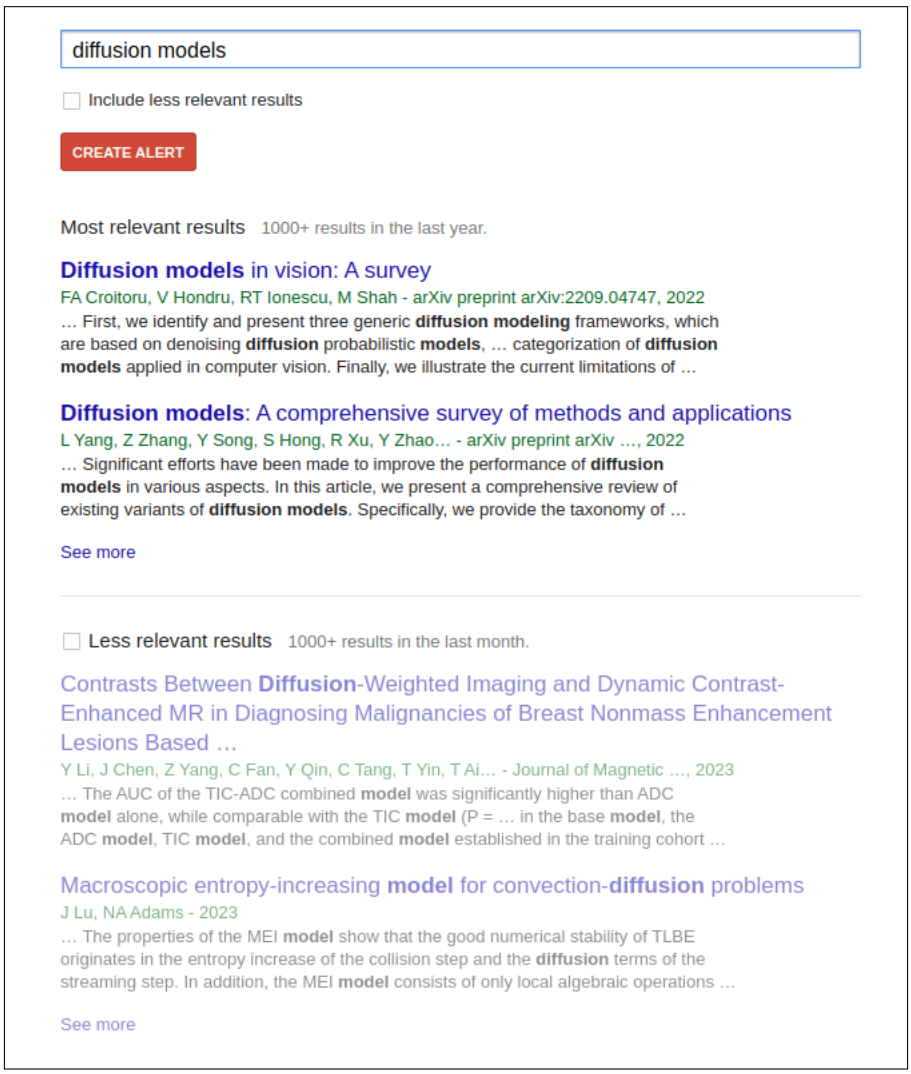


Figure 2.2: Enabling Email Alerts for Google Scholar

Shortcomings

Google Scholar does not provide a method for users to provide labels to the papers it sends in email alerts. Therefore, the email alerts never get to learn from what the user

feedback. As well, Google Scholar does not provide any alternative sources such as Reddit and Twitter.

2.3 Research Rabbit

Research Rabbit [6] is an application that provides users with a way to create collections of papers and from this collection explore related work to delve further into a desired topic. An example of the application is shown in Figure 2.3.

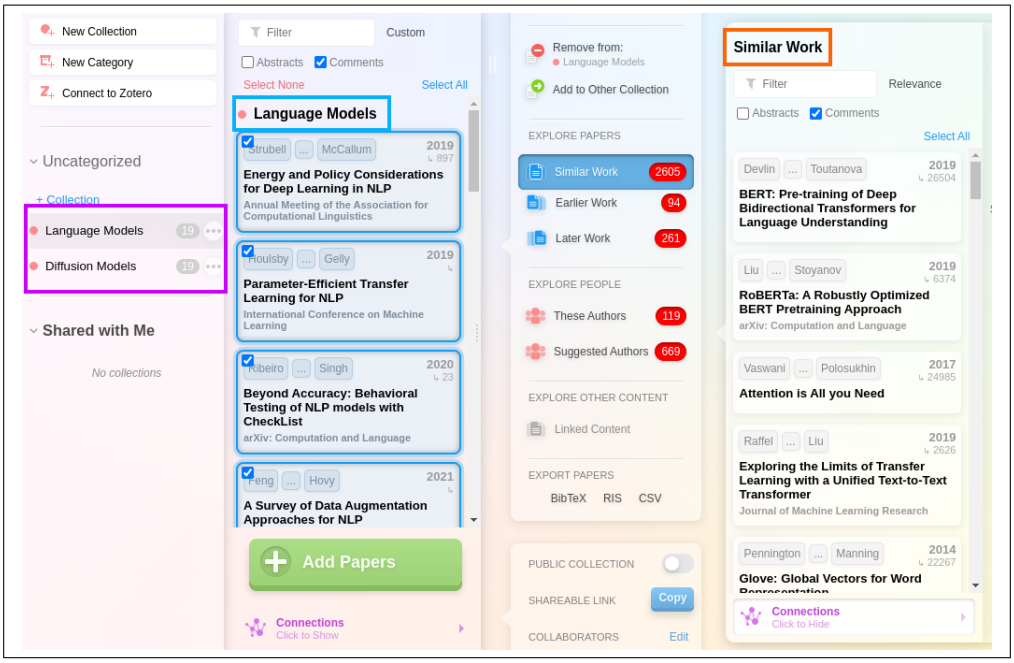


Figure 2.3: The user interface for research rabbit. Outlined is the collections of papers (magenta box), selected collection (cyan box), and work related to the selected collection (orange box).

Shortcomings

While Research Rabbit does provide users with a method to add papers to a collection and view similar work, it doesn't allow them to label these recommendations to improve the system. As well, it only provides papers and no other sources.

2.4 Deeplearn

Deeplearn [7] is a web application that provides users with recent tweets from deep learning scholars, such as Yann LeCun, as well as trending and new papers from arxiv (Figure 2.4). Furthermore, the application allows users to create their own keyword monitors for arxiv.

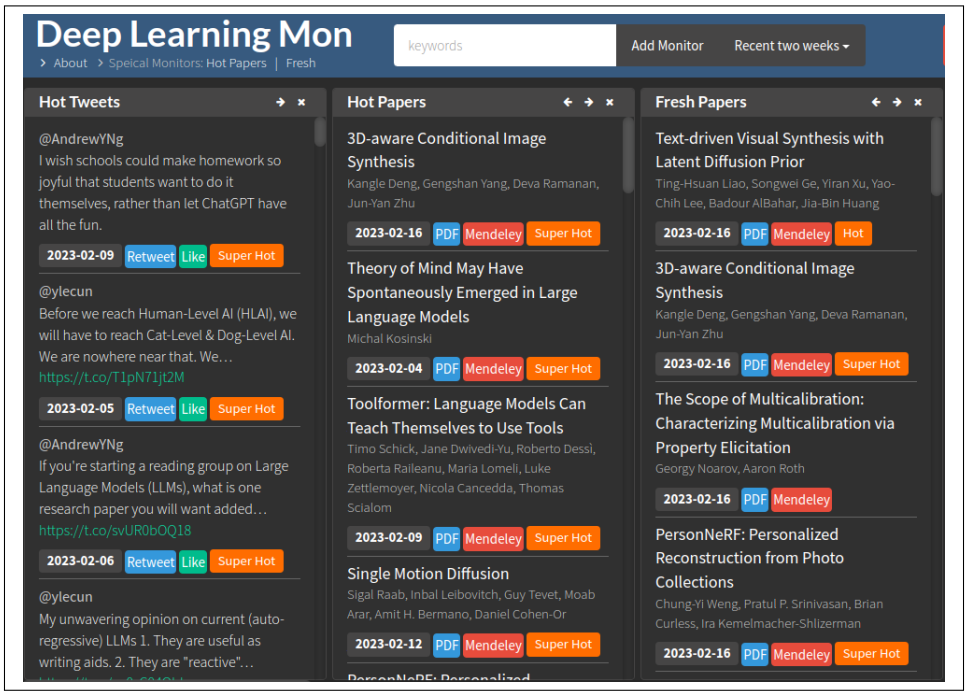


Figure 2.4: User Interface for deeplearn.org

Shortcomings

While deeplearn does provide recommendations based on keywords, it does not allow the user to label any of these recommendations to improve in the future.

2.5 Up2Date Research vs. Related Work

As described in the *shortcomings* section of each of the related works, it is clear there are differing limitations to each work. Figure 2.5 outlines a venn diagram that showcases

what each work provides a given user. By design, Up2Date Research is the only one that provides users with multiple sources, recommendations and allows for user feedback to improve upon said recommendations.

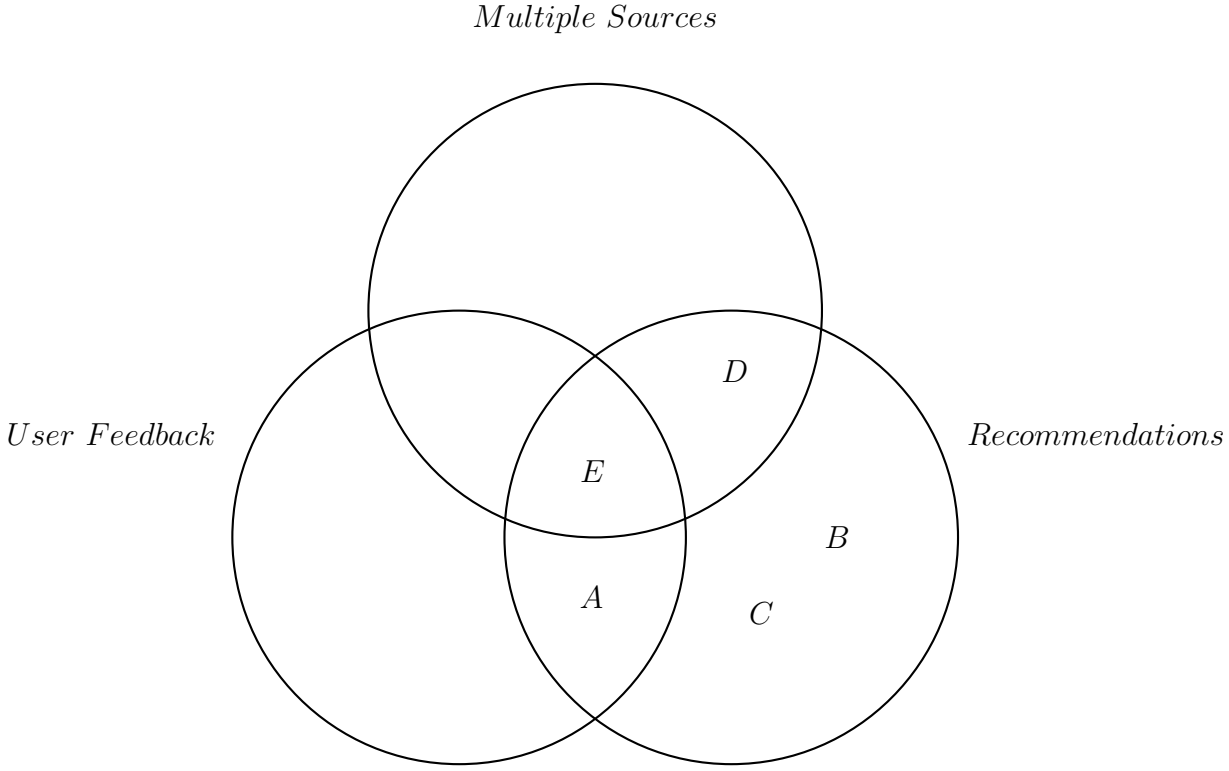


Figure 2.5: Venn Diagram of Related Work and Up2Date Research - Arxiv Sanity (A), Google Scholar (B), Research Rabbit (C), Deeplearn (D) and Up2Date Research (E)
.

Chapter 3

Research Questions

There were a number of research questions that needed to be answered through the implementation of experiments. They were as follows:

1. What Linear Model and Text Vectorizer should be used?
2. Should per source or combined-source training be used?
3. What is the best Metric Performance that can be achieved on the Sample dataset?
4. What is the Metric Performance difference between the Linear and Transformer Models?
5. What is the effect on Metric Performance as training size increases?
6. What is the effect of Classification Threshold on Metric Performance?
7. Do the False Positive and False Negative predictions appear logical?
8. What is the performance on future recommendations based on past labels?

3.1 What Linear Model and Text Vectorizer should be used?

As mentioned previously, there are a total of four linear models and four text encoders. In order to select a linear model and text encoder combination, each combination must be

tested on the proxy dataset for reddit and twitter. Then, the average F1 score between the two datasets will be computed. The highest F1 score of these combinations will be chosen as the best linear model and text encoder to use for further experiments. As previously mentioned, the proxy dataset was used, in lieu of the sample dataset, as the training and test size were much bigger and thus less sensitive to overfitting as compared to the smaller sample dataset size.

3.2 Should per-source or combined-source training be used?

Models will be trained, and provide inference, on individual sources as well as using all sources together. The idea here is to see if predictive performance can be maintained by training and predicting on all sources, as this would save on computational and storage costs to train and store models respectively.

3.3 What Metric Performance can be achieved on the Sample Dataset?

Based on the results from per-source and multi-source training, the best possible metric performance on the sample dataset can be determined.

3.4 What is the Metric Performance difference between the Linear and Transformer Models?

As there is a large computational and storage cost difference between the linear and transformer models, it is useful to know the additional, if any, performance increase for this cost.

3.5 What is the effect on Metric Performance as training size increases?

Currently, we have a sample dataset training size of approximately 650 data points. It would be helpful to know how the metric performance changes as the training size increases and what is a possible upper bound on model performance. This can be accomplished with the use of the proxy dataset.

3.6 What is the effect of Classification Threshold on Metric Performance?

Normally a model classification threshold of 0.5 is used, so that when a prediction is 0.5 or over the prediction is 1 and 0 otherwise. By increasing, or decreasing, this threshold one can change the precision, recall and F1 score values. An analysis on this would be helpful to know if a classification threshold other than 0.5 would be beneficial.

3.7 Do the False Positive and False Negative predictions make sense?

While aggregate metric performance is very useful in assessing how well models compare to each other, a qualitative analysis of the model predictions is helpful to assess if the model is making "adequate" decisions for false positives and false negatives. If not, it is possible the model is leveraging some unseen noise that happens to be a useful signal to this specific training set and test set. This would be undesirable as this "noise" would likely only be found in this isolated training set and not in future data.

3.8 What is the performance on future recommendations based on past labels?

The performance on the sample dataset as detailed in Section 3.3 did not take into account splitting the train and test sets temporally. In this experiment, the plan is to split the train

and test sets based on the time they were labeled; with the test set having been labeled after the train set. This is to check whether past labels are able to accurately predict future recommendations.

Chapter 4

Methods

This chapter describes the experiment framework, models, text vectorizers, types of experiments and metrics used.

4.1 Experiment Framework

In order to facilitate the implementation of multiple experiments, a framework was developed that made use of command line arguments. A list of all of these arguments can be found in Table 4.1.

Argument	**Description**
experiment_type	The type of experiment to perform
training_csv	The data to use to train the model
model	The type of model to use for the experiment
encoder	(Optional - "tfidf" Default) The type of encoder to use. Transformers have their own and thus do not need one.
epochs	(Optional - 4 Default) The number of epochs to use for the neural network models
batch_size	(Optional - 8 Transformer Default) The batch size to use for the neural network models.
alpha	(Optional - 1E-5 Transformer Default) The learning rate to use for the neural network models.

data_splits	(Optional - 5 Default) The number of splits to use for running the "increasing training size" experiment.
vector_size	(Optional - 300 Default) The vector size to use for the text vectorizer.
test_percent	(Optional - 0.15 Default) The percent of training data to use as the test set. If a test csv is provided then this is not applied.
classifier_threshold	(Optional - 0.5 Default) Threshold at which to classify a data point as 1 or 0.
resample_rare_class	(Optional - False Default) Whether or not to resample the minority class. A naive resampling method is used by just resampling without replacement and appending onto the existing training data.
save_model_checkpoint	(Optional - False Default) Whether or not to save the trained model checkpoint. Neural networks only.
save_raw_predictions	(Optional - False Default) Whether or not to save the raw predictions from the model.
test_csv	(Optional) The csv file to use as the test set. If no file is provided, a portion (‘test_percent‘) of the training_csv is used.
checkpoint_metric	(Optional: F1 Default) The metric for which to store the best performing model across epochs. Only used for neural networks.
optimizer	(Optional - AdamW Default) The optimizer to use for the neural networks. Either AdamW or AdaFactor
betas	(Optional - [0.9, 0.999] Default) The betas to use for the AdamW optimizer
eps	(Optional - 1E-8 Default) eps to use for the AdamW optimizer
weight_decay	(Optional - 1E-2 Default) Weight decay for the AdamW and AdaFactor optimizer
clip_threshold	(Optional - 1.0 Default) Clip threshold for the AdaFactor optimizer
beta1	(Optional - None Default) beta1 for the AdaFactor optimizer
scale_parameter	(Optional - True Default) Whether to use scale_parameter for the AdaFactor optimizer
relative_step	(Optional - True Default) Whether to use relative_step for the AdaFactor optimizer
warmup_init	(Optional - False Default) Whether to use warmup_init for the AdaFactor optimizer

n_datapoints	(Optional - All Data Points Used Default) The number of data points to use from the provided training csv file

Table 4.1: Experiment Framework Arguments

There were two types of models that were used: linear models and neural networks. Depending on the model and type of experiment to run, some of these arguments are unnecessary. It is up to the user to determine when to pass in an argument or whether to just use a default value.

4.2 Models

As mentioned previously, two types of models are used including linear and neural network models. One type of neural network model was used, distilBERT, which is a transformer model. Four types of linear models were used including support-vector machines, logistic regression, XGBoost, and Naive Bayes.

For the neural networks, the hyperparameters can be set via the command line for the experiment. For the linear models, a k-fold cross validation is always performed with the training data first with the use of a hyper-parameter grid search. In summary the five models used in this project are as follows:

1. Transformer DistilBERT
2. XGBoost
3. Logistic Regression
4. SVM
5. Naive Bayes

Transformer DistilBERT

DistilBERT [8] has the same general architecture as BERT [9] except that the token-type embeddings and the pooler are removed while the number of layers is reduced by a factor of 2.

In order to train DistilBERT, a method called "Knowledge Distillation" is used in which a student, DistilBERT in this case, is trained to reproduce the behavior of the teacher, BERT. The complete loss function used in this training includes the linear combination of the (1) Supervised Training Loss, (2) Distillation Loss and (3) Cosine Embedding Loss.

With these changes, DistilBERT still retains 97% of BERT's performance while only having 60% as many parameters.

XGBoost

XGBoost [10] is based on a decision tree. A tree ensemble method uses K additive functions to predict the output (Equation 4.1).

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^{K} f_k(x_i), f_k \in F \tag{4.1}$$

Where $F = f(x) = w_{q(x)}(q : \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T)$ is the space of regression trees. q represents the structure of each tree that maps an example to the corresponding leaf index. T is the number of leaves in the tree. f_k corresponds to an independent tree structure q with leaf weights w. The score given to an input is the sum of the output from each tree. In order to learn the set of functions used in the model, the authors minimize the regularized objective function (Equation 4.2).

$$\mathbb{L}(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \tag{4.2}$$

$$where\ \Omega(f) = \gamma\mathbb{T} + \tfrac{1}{2}\lambda\|w\|^2$$

l, which is differentiable and convex, measures the distance between the prediction $\hat{y}_i$ and y_i. The second term Omega penalizes the complexity of the model with γ and λ being user chosen constants.

Logistic Regression

Logistic Regression is a linear model for classification [11]. The cost function associated with this model uses Elastic-Net regularization (Equation 4.3).

$$\min_{w,c} \frac{1-\rho}{2} w^T w + \rho \|w\|_1 + C \sum_{i=1}^{n} \log(\exp(-y_i(X_i^T w + c)) + 1) \tag{4.3}$$

w is the learned parameter and ρ is the mixture of the l_1 and l_2 regularization parameters. Elastic-Net is equivalent to to l_1 when $\rho = 1$ and l_2 when $\rho = 0$. C scales the effect of regularization.

SVM

Support Vector Machines [12] are a set of supervised learning methods used for classification, regression and outlier detection. SVMs are unique in the fact that they solve an optimization problem involving the dual which utilizes the inner product between data points rather than the data points themselves. This results in SVMs being very effective in high dimensional spaces but can be slow when the number of data points is very large.

Given training vectors $x_i \in \mathbb{R}^p, i = 1, .., n$ and $y \in \{1, -1\}^n$, the goal is to find $w \in \mathbb{R}^p$ and $b \in \mathbb{R}$ such that the prediction given by $sign(w^T \phi(x) + b)$ is correct for most samples. SVC solves the optimization problem in Equation 4.4.

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i \tag{4.4}$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i$$

$$\zeta_i \geq 0, \ i = 1, ..., n$$

The dual problem is outlined in Equation 4.5.

$$\min_{\alpha} \frac{1}{2}\alpha^T Q\alpha - e^T\alpha \tag{4.5}$$

$$\text{subject to } 0 \leq \alpha_i \leq C,\ i = 1, ..., n$$

Where e is the vector of all ones and Q is an n *by* n positive semi-definite matrix. The terms a_i are the dual coefficients we are solving for in the dual problem. Each Q_{ij} can be found using Equation 4.6.

$$Q_{ij} = y_i y_j K(x_i, x_j) \tag{4.6}$$

There is also the kernel, K, which is computed according to Equation 4.7.

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j) \tag{4.7}$$

Once the optimization problem is solved, the output of the decision function, for a given sample, is determined using Equation 4.8.

$$\sum_{i \in \mathbb{SV}} y_i \alpha_i K(x_i, x) + b \tag{4.8}$$

SVM compute and storage requirements increase rapidly with the number of training vectors.

Naive Bayes

Naive Bayes [13] is a method of applying Bayes' theorem with the "naive" assumption of conditional independence between every pair of features given the value of the class variable (Equation 4.9).

$$P(x_i|y, x_1, ..., x_{i-1}, x_{i+1}, ..., x_n) = P(x_i|y) \tag{4.9}$$

$$P(y|x_1, ..., x_n) = \frac{P(y)P(x_1, ..., x_n|y)}{P(x_1, ..., x_n)} \tag{4.10}$$

Using the naive conditional independence assumption of Equation 4.9 and inserting it into Equation 4.10, we receive Equation 4.11.

$$P(y|x_1, ..., x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, ..., x_n)} \tag{4.11}$$

$$P(y|x_1, ..., x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

The predicted y value can be found by Equation 4.12.

$$\hat{y} = \arg\max_{y} P(y)\prod_{i=1}^{n} P(x_i|y) \tag{4.12}$$

4.3 Text Encoders

There are four tokenization methods that were used in order to vectorize text to be passed into the machine learning models. They are as follows:

1. TFIDF
2. TFIDF-GloVe
3. Universal Sentence Encoder
4. Bi-Gram

TFIDF

Term frequency inverse document frequency (TF-IDF) is a method used to convert a string of text into a numerical vector [14]. Term Frequency (TF) is calculated as follows as in Equation 4.13.

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \tag{4.13}$$

The numerator $f_{t,d}$ is the raw count of the word t in the document d, while the denominator is the total number of (non-unique) terms in the document d. Inverse Document Frequency (IDF) is a measure of how much information a word provides. For example if it is common across all documents then the value will be low, while rarer words across all documents will be high. It is calculated according to Equation 4.14.

$$idf(t,D) = \log \frac{N}{|d \in D : t \in d|} \tag{4.14}$$

N is the total number of documents in the corpus $N = |D|$ while the denominator is the total number of documents in which the word appears at least once. Thus tf-idf is defined in Equation 4.15.

$$tfidf(t,d,D) = tf(t,d) * idf(t,D) \tag{4.15}$$

A vector for a given paragraph can then be constructed by selecting all or a subsection of the tfidf values for each term in the paragraph.

GloVe

GloVe is a pre-trained hashmap of vectors for a given token [15]. GloVe stands for Global Vectors because in this model, the global corpus statistics are captured. In order to begin to explain the process for attaining the GloVe word embeddings, we start with a word2word co-occurrence matrix denoted by X whose entries X_{ij} tabulate the number of times word j occurs in the context of word i.

$$X_i = \sum_k X_{ik} \tag{4.16}$$

X_i, in Equation 4.16, is the number of times any word appears in the context of word i

$$P_{ij} = P(j|i) = \frac{X_{ij}}{X_i} \tag{4.17}$$

P_{ij}, Equation 4.17, is the probability of word j to appear in the context of word i. GloVe takes the approach of learning word embeddings from the ratio of co-occurrence probabilities rather than the probabilities themselves. The most general model takes the form of Equation 4.18.

$$F(w_i, w_j, \widetilde{w_k}) = \frac{P_{ik}}{P_{jk}} \tag{4.18}$$

$w_i, w_j \in \mathbb{R}^d$ are word vectors and $\widetilde{w_k} \in \mathbb{R}^d$ are context vectors. The right-hand side of this equation is extracted from the corpus while the left hand side will be explained further. The authors wanted F to encode the information present in the ratio $\frac{P_{ik}}{P_{jk}}$. As vector spaces are linear structures, a natural way to do this is using vector differences. With this in mind, F can be restricted to use only the difference of the two target word vectors (Equation 4.19).

$$F(w_i - w_j, \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}} \tag{4.19}$$

The function is further reduced to taking the inner product between the vector difference and the context vector (Figure 4.20).

$$F((w_i - w_j)^T \widetilde{w}_k) = \frac{P_{ik}}{P_{jk}} \tag{4.20}$$

This prevents F from mixing the vector dimensions in undesirable ways. The distinction between a word vector and a context word vector is arbitrary and thus the two should be interchangeable. To do so consistently, we must not only be able to interchange $w <-> \widetilde{w}$ but also $X <-> X^T$. The final model should be invariant under this relabeling. To restore this symmetry we must require that F be a homomorphism between the groups $(R, +)$ and $(R^+, \times)$, as shown in Equation 4.21.

$$F((w_i - w_j)^T \widetilde{w}_k) = \frac{F(w_i^T \widetilde{w}_k)}{F(w_j^T w_k)} \tag{4.21}$$

The solution to Equation 4.21 for F is in fact the exponent function, $F = e$.

$$F((w_i^T \widetilde{w}_k)) = P_{ik} = \frac{X_i k}{X_i} \tag{4.22}$$

$$w_i^T \widetilde{w}_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i) \tag{4.23}$$

Equation 4.23 would exhibit the exchange symmetry if not for the $\log(X_i)$; however, this term is independent of k so it can be absorbed into a bias b_i for w_i. Finally, adding an additional bias $\widetilde{b}_k$ for $\widetilde{w}_k$ restores the symmetry.

$$w_i^T \widetilde{w}_k + b_i + \widetilde{b}_k = \log(X_{ik}) \tag{4.24}$$

The loss function to optimize for, Equation 4.25 which is based on word vectors, is an altered least squares regression problem with a weighting function $f(X_{ij})$.

$$\text{Loss} = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widetilde{w}_j + b_i + \widetilde{b}_j - \log(X_{ij}))^2 \tag{4.25}$$

$$f(x) = \begin{cases} \frac{x}{x_{\max}^{\alpha}} \; if \; x < x_{\max} \\ 1 \text{ otherwise} \end{cases} \tag{4.26}$$

In their experiments, $x_{\max}$ of 100 and alpha of $\frac{3}{4}$ was used. Once a corpora was chosen by the authors, it is tokenized using the Stanford tokenizer [16] and a co-occurrence counts matrix X is constructed. A context window of 10 to the left and right of the current word i is used. A weighting is also applied to this given how far away the context word is from the chosen word. In this way, words that are more distant do not count as much as those that are closer. A matrix W and $\overline{W}$ are both randomly initialized and then updated using the AdaGrad optimizer. When X is symmetric, W and $\overline{W}$ only differ due to random initializations; the two vectors should perform equivalently.

As GloVe only provides us with a vector for each word, and thus a matrix for the entire paragraph of text, a vector for the input paragraph is obtained by reducing across the paragraph token size dimension by computing the mean (Equation 4.27).

$$V_{\text{sentence}} = \frac{1}{|S|} \sum_{w \in S} GloVe(w) \tag{4.27}$$

TFIDF GloVe

In order to apply tfidf-glove to a paragraph, the normalized tfidf for each token is first determined (Equation 4.28).

$$tfidf_{norm}(w) = \frac{tfidf(w)}{\sum_{w \in S} tfidf(w)} \tag{4.28}$$

Where $w \in S$, denotes each word in the input paragraph S. Then the weighted average, with respect to the normalized tfidf, is computed across all tokens in the paragraph:

$$V_{sentence} = \sum_{w \in S} tfidf_{norm}(w) * GloVe(w) \tag{4.29}$$

Universal Sentence Encoder

The universal sentence encoder encodes a sentence into a vector using a Deep Averaging Network (DAN) [17]. Input embeddings for words and bi-grams are first averaged together and then passed through a feedforward deep neural network to produce sentence embeddings. The DAN takes as input a Penn Treebank Tokenization (PTB) string and outputs a 512 dimensional vector. The DAN is trained on several downstream tasks such as Skip-Thought for the unsupervised learning from arbitrary running text, a conversation input-response task for the inclusion of parsed conversational data and classification tasks for training on supervised data.

Bi-Gram

A bi-gram is calculated by creating a one-hot encoding of pairs of words [14]. These pairs of words are generated on a rolling window basis for the text. As an example, "Hello how are you" would be transformed into bi-grams as "hello how", "how are" and "are you". Bi-grams help provide the conditional probability of a word appearing given the occurrence of a preceding word. This conditional probability is outlined in Equation 4.30.

$$P(W_n|W_{n-1}) = \frac{P(W_{n-1}, W_n)}{P(W_{n-1})} \tag{4.30}$$

Where $P(W_{n-1}, W_n)$ is the number of times the word W_{n-1} and W_n appear together in the corpus and $P(W_{n-1})$ is the number of times W_{n-1} appears in the corpus. A vector is then created by selecting all or a subsection of all bi-gram probabilities.

4.4 Experiment Types

Two different types of experiments were made available using this framework including:

- run_model
- run_increasing_training_set

'run_model' simply trains the model on a (subset) of the dataset, using cross validation on a hyper-parameter grid if it is a linear model, and is then applied to the test set. 'run_increasing_training_set' is used to get an understanding of model performance as the training set increases. For each chosen training set size, 'run_model' is performed and the results are recorded for that unique number of training data points.

4.5 Metrics

As this is a binary classification task, there are three major metrics that are commonly used to assess the performance of a model's prediction. They are recall, precision and F1 score [18]. Recall is the ratio of true positives over true labels (Equation 4.31).

$$Recall = \frac{TP}{(TP + FN)} \tag{4.31}$$

Precision is the ratio of true positives out of all positives the model predicts (Equation 4.32).

$$Precision = \frac{TP}{TP + FP} \tag{4.32}$$

F1 Score is a combination of Recall and Precision (Equation 4.33).

$$F1\ Score = \frac{2 * (Precision * Recall)}{Precision + Recall} \tag{4.33}$$

As we would like to maximize both recall and precision, the F1 Score will be used as the default metric for all experiments.

Chapter 5

Evaluation

This chapter details all of the experiments conducted to answer the research questions proposed in Chapter 3.

5.1 Choosing a Linear Model and Text Vectorizer

This experiment aims to answer the research question "What Linear Model and Text Vectorizer should be used?" (Section 3.1).

In order to choose an appropriate linear model and text vectorizer, the sixteen combinations of the four linear models and four vectorizers were tested on proxy reddit and twitter datasets which consist of approximately fifteen thousand rows of data each. These proxy dataset sets were used as they were less likely to suffer from the noise of the much smaller sample dataset. While many of the top performing models on text classification benchmarks are neural networks [19], on some datasets such as 20News and Amharic News Text Classification, the best performing models are SVM + TFIDF and Naive Bayes + TFIDF respectively. While we could just choose to use one of these model and encoder combinations, it may prove useful to conduct this experiment to see if there are different results.

The average F1 score grouped by encoder and model across the sources of reddit and twitter in the proxy dataset was computed and ordered from largest (top) to smallest (bottom) in Figure 5.1. The average F1 score is the column "avg_f1_score" and the magnitude of difference between the two scores can be found in the column "f1_score_diff". "avg_f1_score" gives the average performance of the model-encoder combination across both reddit and

twitter from the proxy dataset, while "f1_score_diff" gives the spread of this average. Ideally, we would want to maximize "avg_f1_score" and minimize "f1_score_diff". The two SVM models appear to be in the top three with the encoders being universal sentence encoder and tfidf. For this reason, it seems reasonable to use SVM as the model. However, some more thought must be put into what encoder to use.

model	encoder	avg_f1_score	f1_score_diff
svm	universal_sentence_encoder	0.842088	0.082529
xgb	universal_sentence_encoder	0.836534	0.089537
svm	tfidf	0.820886	0.043197
logistic_regression	tfidf	0.820144	0.051958
	universal_sentence_encoder	0.819977	0.113757
naive-bayes	universal_sentence_encoder	0.794855	0.129231
xgb	tfidf	0.791337	0.059093
svm	tfidf-glove	0.771547	0.043761
xgb	tfidf-glove	0.768091	0.044525
naive-bayes	tfidf-glove	0.762584	0.008178
logistic_regression	tfidf-glove	0.749517	0.070330
naive-bayes	tfidf	0.687521	0.225286
logistic_regression	bi-gram	0.437268	0.091982
svm	bi-gram	0.423495	0.148557
xgb	bi-gram	0.360850	0.115845
naive-bayes	bi-gram	0.316052	0.205105

Figure 5.1: Linear Model and Encoder Combination Results Averages

There are two main things to consider when analyzing the two encoders. The processing times and the tokenization output.

Encoder Processing Times

Figure 5.2 shows the processing time for encoding an increasing training corpus size with tfidf. We can see that it increases linearly with the training size.



Figure 5.2: TFIDF Encoder Times

Similarly, Figure 5.3 shows the processing time for encoding an increasing corpus size with universal sentence encoder.

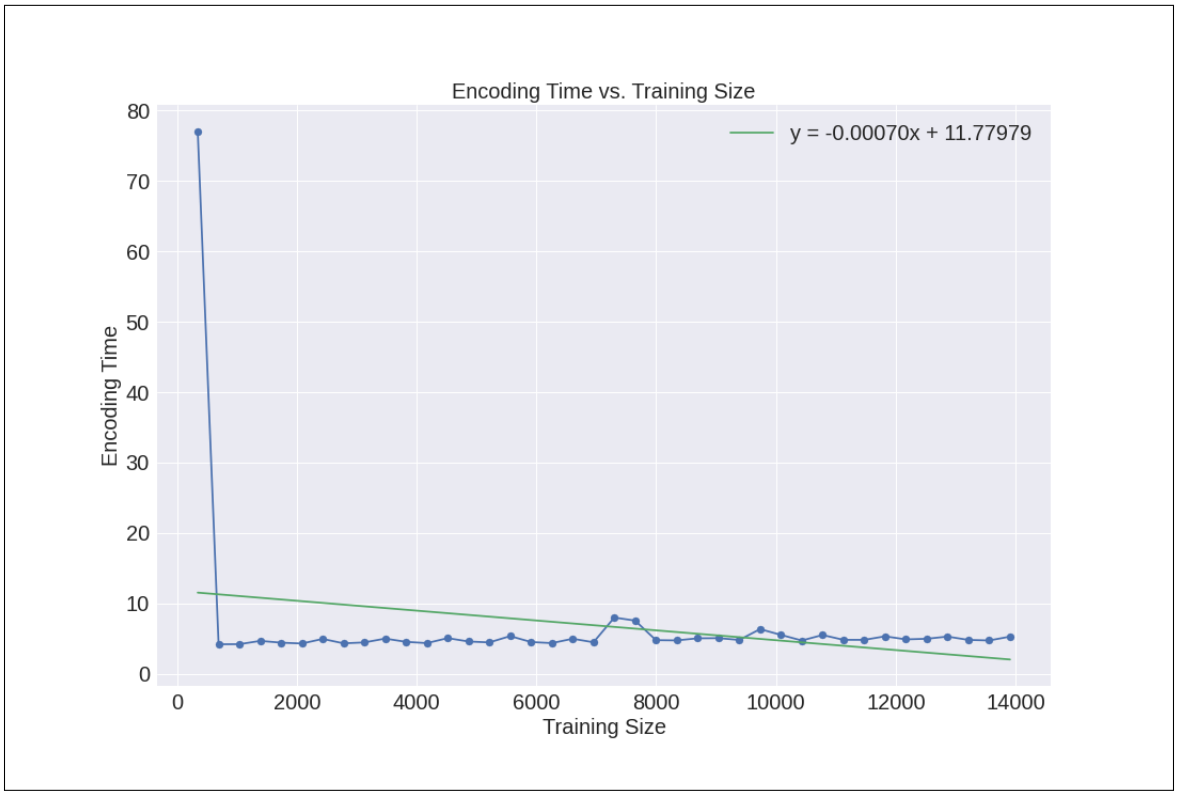


Figure 5.3: Universal Sentence Encoder Times

This may not make much sense at first glance, but the reason for the initial long processing time is due to the encoder first being loaded into memory and then being initialized (as it is a saved model). Once loaded into memory we can analyze the performance in Figure 5.4.

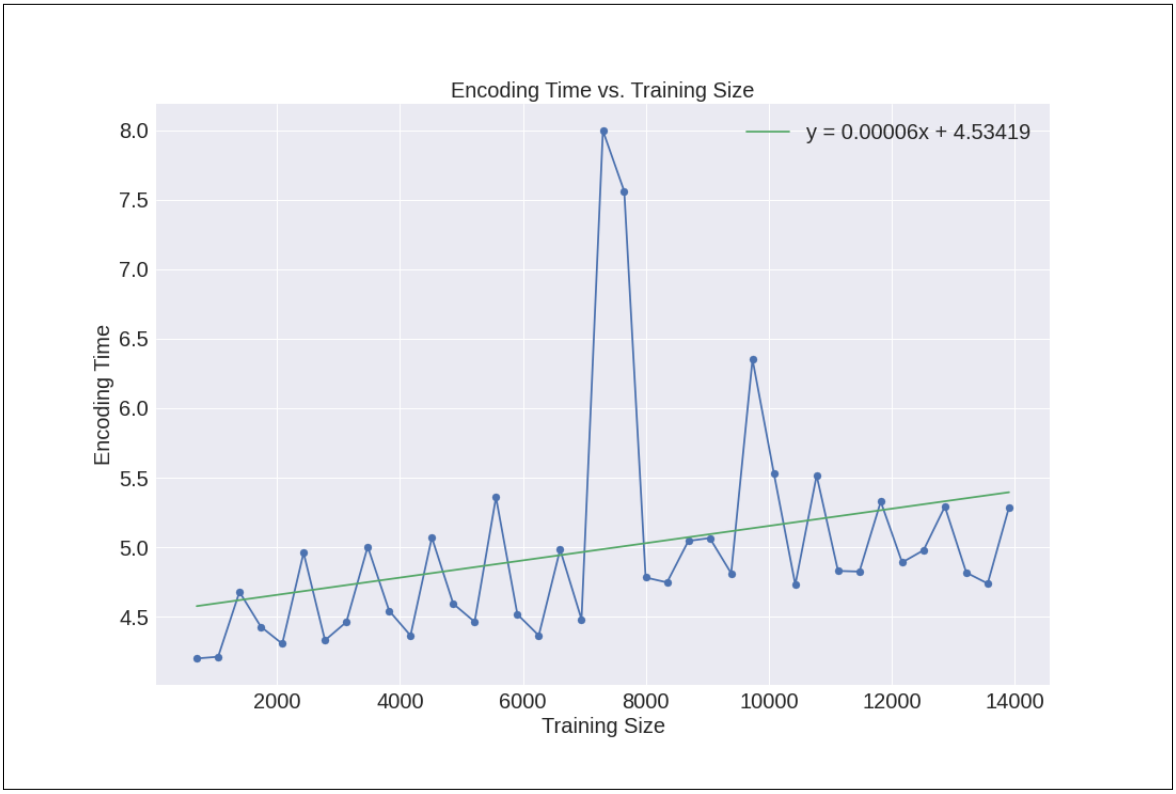


Figure 5.4: Universal Sentence Encoder Times Without Start

Here we can see that there is also a linear increase in the processing time for the training corpus.

Conclusion

There does not appear to be a large trade-off between the encoder processing times for both the universal sentence encoder and tfidf as universal sentence encoder only has to be loaded once in memory before being applied to all required training corpuses.

Encoder Tokenization

Let's say we had the following corpus of words to vectorize:

“graph neural network gnn relu equivariant convolution gradient neuron hessian perceptron”

These all seem important to the research interests of Kimon and thus we would want the tokenization to be as close to these root words as possible to ensure the underlying meaning isn't changed. Let's see how each of the encoders tokenizes the corpus:

TFIDF

['graph', 'neural', 'network', 'gnn', 'relu', 'equivariant', 'convolution', 'gradient', 'neuron', 'hessian', 'perceptron']

For the most part here, it seems as though TFIDF keeps each word intact and does not reduce each word into a sub-word that could be misconstrued as something else.

Universal Sentence Encoder

['graph', 'neural', 'network', 'g', 'nn', 're', 'lu', 'equi', 'variant', 'con', 'volution', 'gradient', 'neuron', 'he', 's', 's', 'ian', 'percep', 'tron']

Universal Sentence Encoder keeps some words such as “graph”, “neural” and “network” intact but splits up words like “gnn” and “hessian” into subwords as these words were not present in its training corpus.

Conclusion

Since Universal Sentence Encoder does not have certain words in its training corpus, it tokenizes them into sub-words which could result in a misconstrued interpretation. For example, “relu” decomposes into “re” and “lu”. Another word such as “receive” may also be decomposed into “re” + “ce” + “ive” in which case the first decomposition result is the same for both, but they are completely different words. For this reason it seems that tfidf would result in better tokenization as it does not have any predefined tokenization corpus and could adapt better to new words and acronyms in the fast moving field of machine learning.

Research Questions Answered

Q. What Linear Model and Text Vectorizer should be used?

A. SVM and tfidf

5.2 Model Performance on the Sample Dataset

This experiment seeks to answer the research questions "Should per-source or combined-source training be used?" from Section 3.2 and "What Metric Performance can be achieved on the Sample dataset?" from Section 3.3.

With the selection of the linear model and its encoder, we are now ready to see the performance of the linear and transformer models on the sample dataset labeled by Kimon. There are a total of three sources in this dataset which include twitter, reddit, and arxiv & pmlr. Each model was trained on each individual source and on all sources together.

Linear Model

Figure 5.5 is the performance of the linear model (SVM with tfidf). Red is the per source trained model (one model for each source) while green is the model that is trained on all sources.

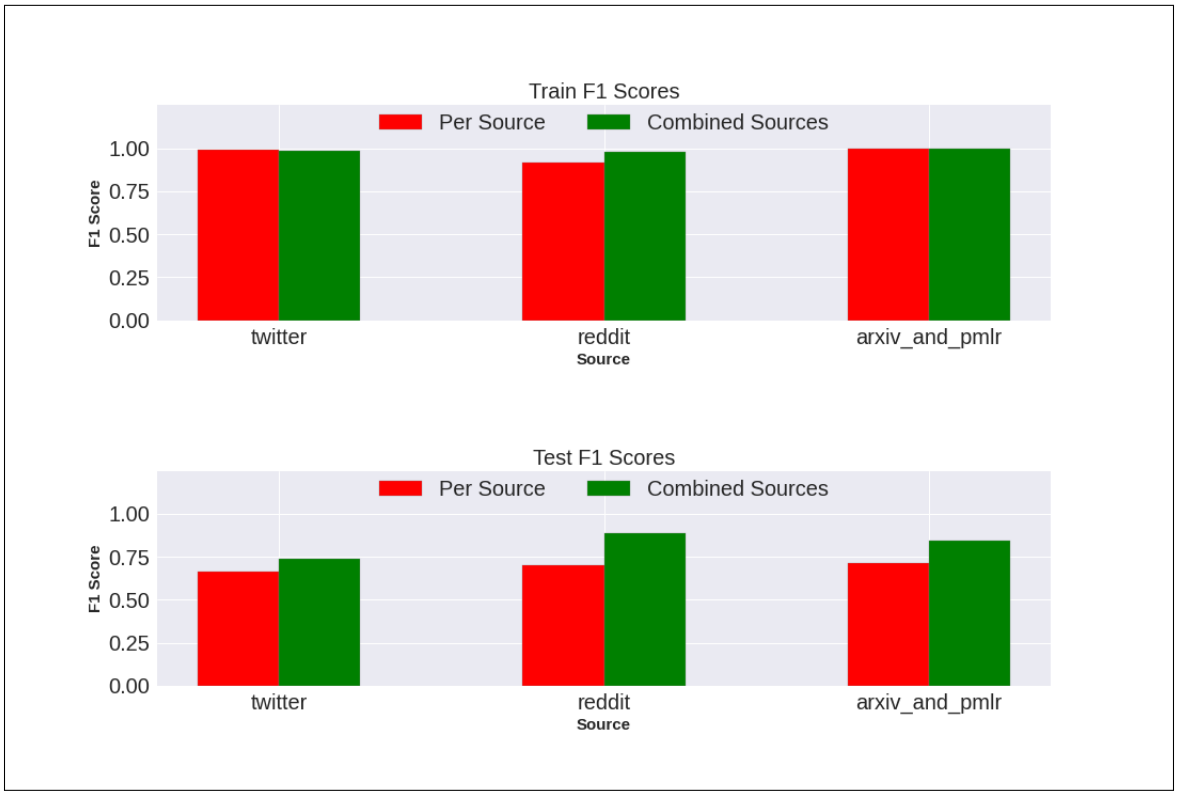


Figure 5.5: Combined Source Predictions for Linear Model

We can see that the combined sources trained model results in equal or better performance across all sources for both the training and test sets. It seems as though the increase in training data is helpful.

Transformer Model

Similarly, we can see in Figure 5.6 that the Transformer model also performs better when trained on all sources (except when predicting on reddit).



Figure 5.6: Combined Source Predictions for Transformer Model

Transformer and Linear Test F1 Scores using Combined Source Training

Comparing the linear and transformer Test F1 Scores for the combined source, we can see that the averages (horizontal lines in Figure 5.7) are nearly equal; however, there are performance differences between twitter and reddit for the two models.

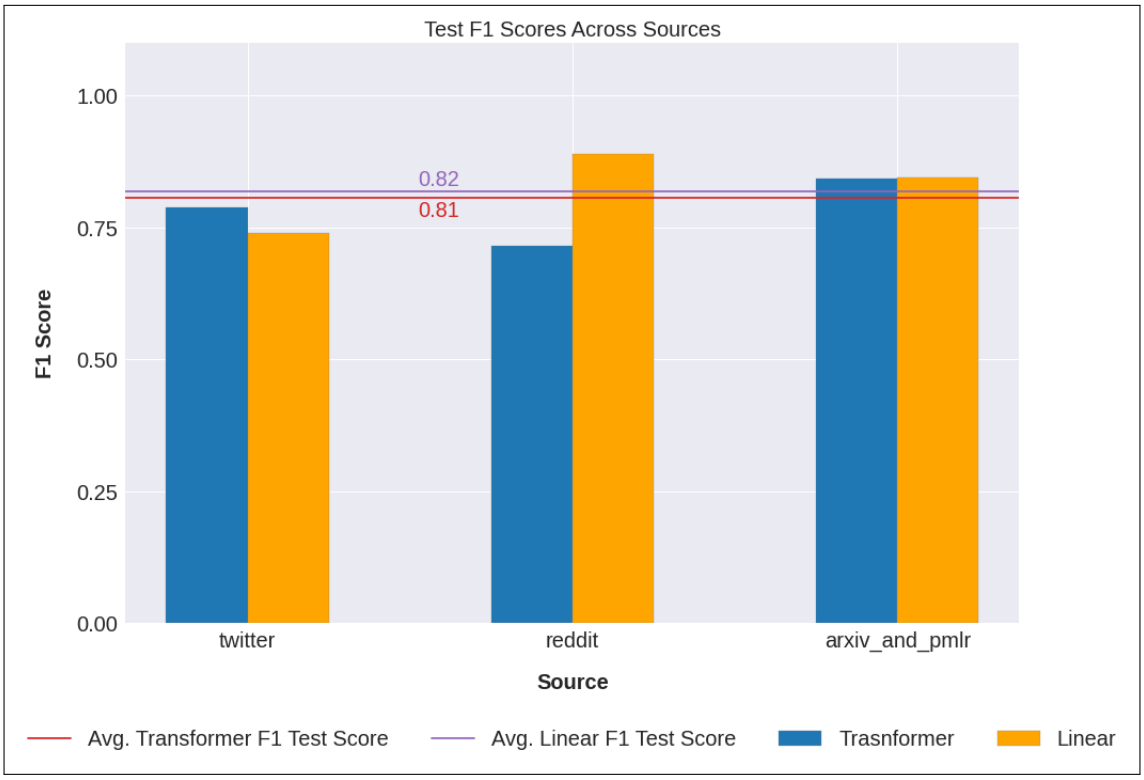


Figure 5.7: Combined Source Predictions Comparisons

Therefore, for this amount of training data (647 data-points), there isn't much of a performance difference between the two models; thus, it would be wise to choose the simpler SVM over the transformer for this reason.

Research Questions Answered

Q. Should per source or combined-source training be used?

A. Combined-source training should be used as it produces a higher test and train metric score.

Q. What Metric Performance can be achieved on the Sample dataset?

A. A Test F1 Score of 82% can be achieved.

5.3 Model Performance on Increasing Training Sizes

This experiment aims to answer the research questions "What is the Metric Performance difference between the Linear and Transformer Models?" from Section 3.4 and "What is the effect on Metric Performance as training size increases?" from Section 3.5. To attain a better understanding of model performance as a function of training size, the use of a proxy dataset was leveraged. The proxy dataset contained labels representing sentiment classification (for reddit and twitter) and category classification (for arxiv). The actual performance achieved on these proxy datasets is not to be used as an exact estimate of future performance for the sample dataset (as their tasks are completely different) but it can give us an idea the difference in performance between the two models as well as an upper bound of performance given the training sizes. Both the transformer and linear model were trained on an increasing training set size. First, these models were trained on less than 2000 data points to be in a similar range to the size of Kimon's sample dataset of 647 data points (Figure 5.8).

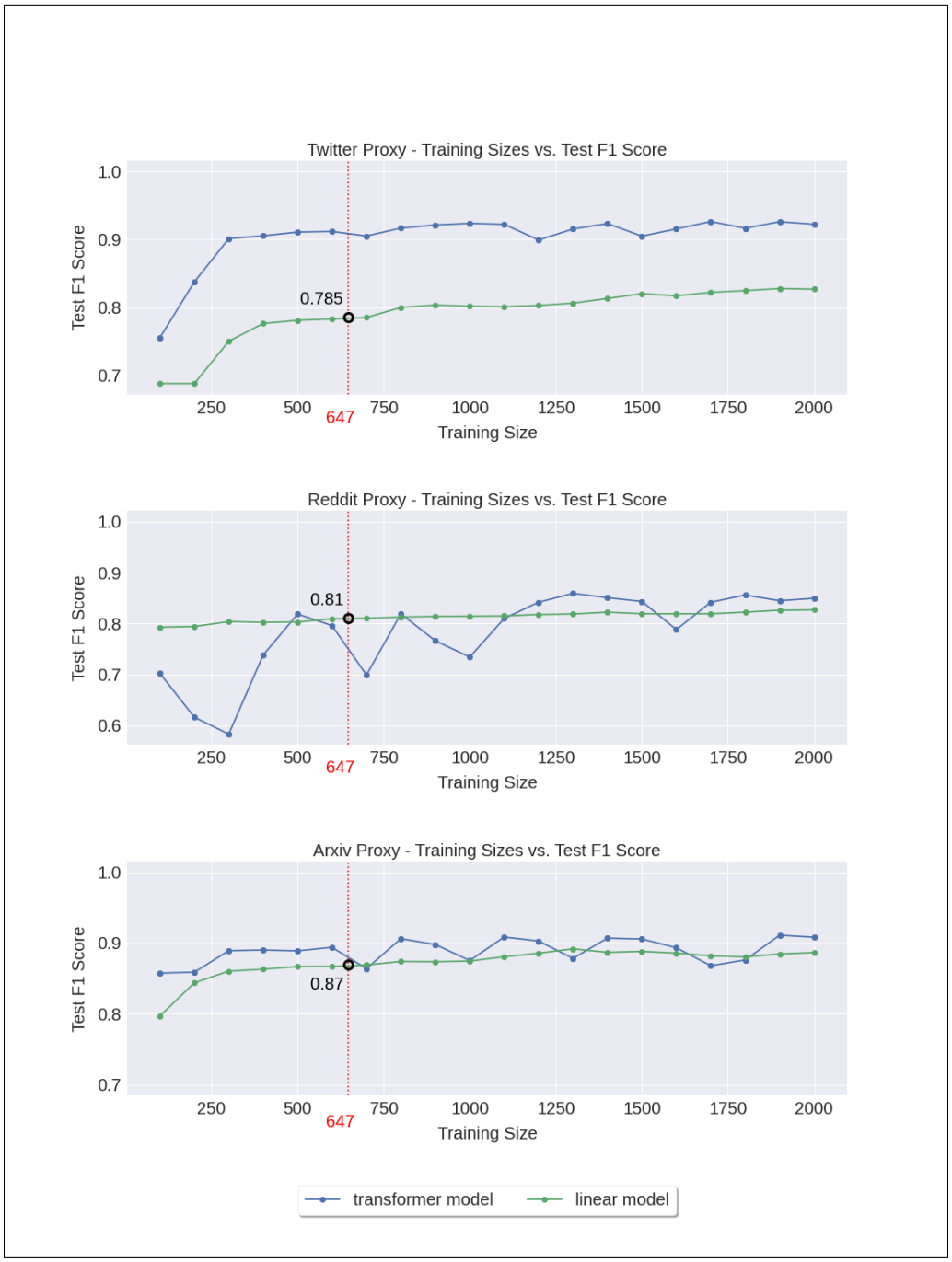


Figure 5.8: Increasing Train Size Transformer and Linear - 2000 data points

Aside from twitter, the differences between model performance for these training sizes seems to be about equal. With an average over the three datasets of 82%. This 82% is very close to the results we have from prediction on Kimon's sample dataset from Figure 5.7. Figure 5.9 details the *difference* between the transformer and the linear model F1 scores. The red bar shows the average score across all data points. Averaging these values across the 3 datasets results in a performance difference of 3.3%.



Figure 5.9: Increasing Train Size Transformer and Linear Difference - 2000 data points

Figure 5.10 shows the performance of both the transformer and linear model on the three sources of the proxy dataset with training sizes ranging from 1000 to 15000 data points. When looking at the performance differences between the transformer and linear model for this range of training data points (Figure 5.11), the transformer maintains a 10%, 11%, and 2% higher Test F1 Score for Twitter, Reddit, and Arxiv respectively over the linear model. An average of these 3 values yield a result of 7.7%. Thus, it would seem as though using a transformer over a linear model with training sizes over 2000 data points would yield a significant performance improvement of about 8% in aggregate over all sources.

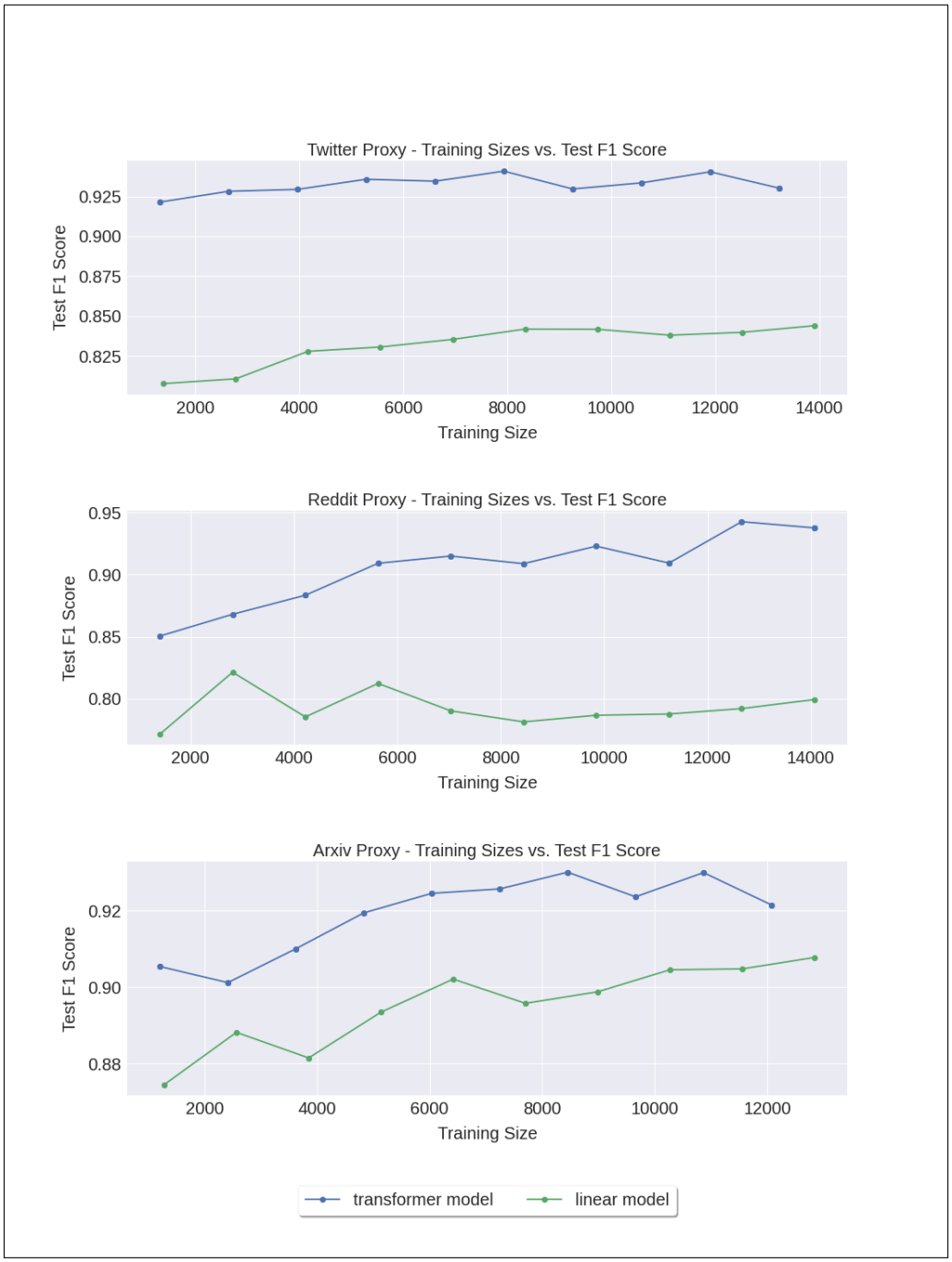


Figure 5.10: Increasing Train Size Transformer and Linear - All data points

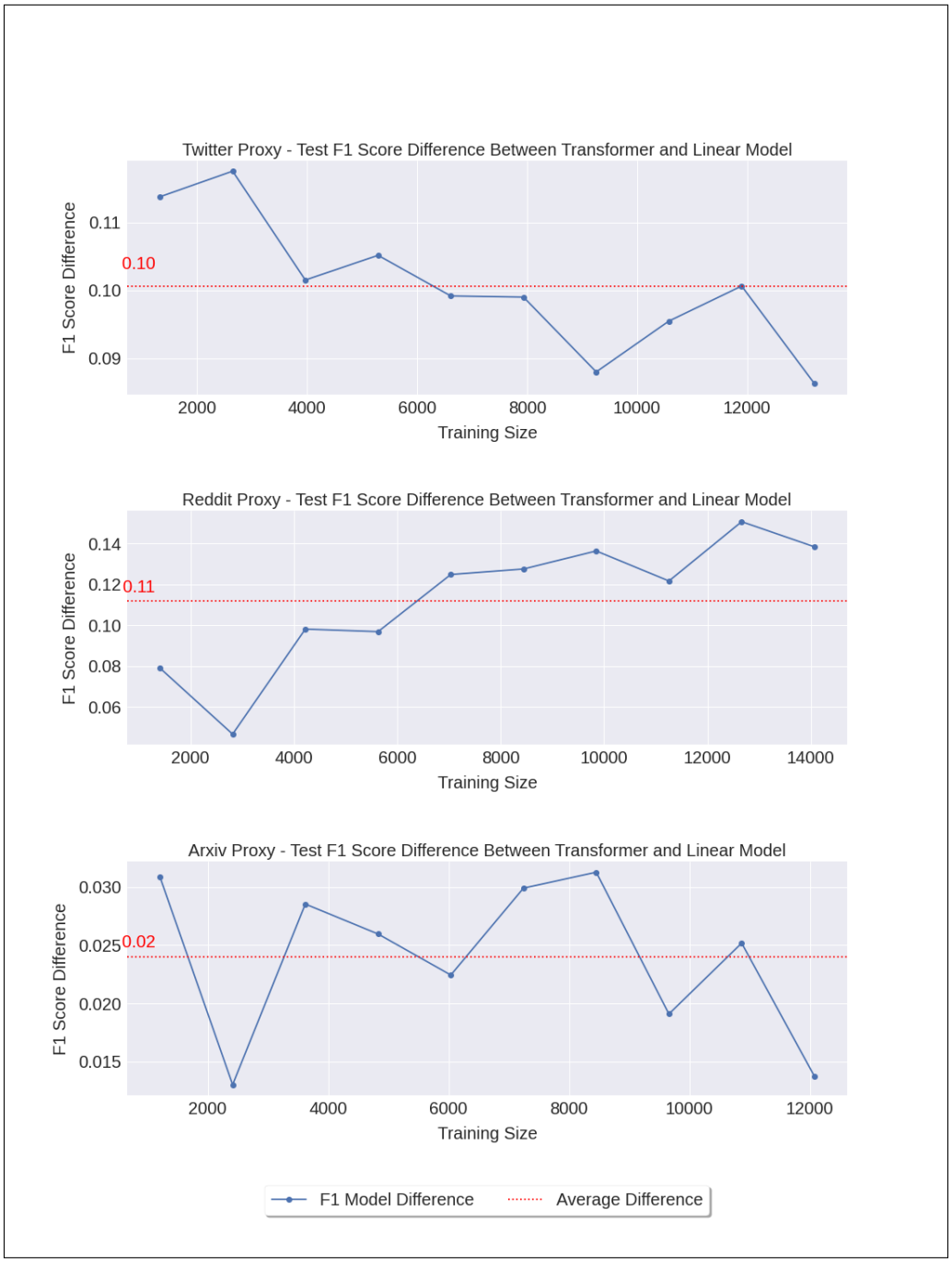


Figure 5.11: Increasing Train Size Transformer and Linear Difference - All data points

Another important factor to analyze is the effect on the train and test F1 Score as a function of increasing training size. Figure 5.12 and 5.13 show the train and test metrics as a function of increasing training size, on each of the three proxy sources, for both the transformer and linear model respectively. For the transformer model (Figure 5.12, we can see that there is still a gap between the train and test F1 Score for each source. This means that the model has overfit to the training data and isn't able to generalize as well to the test data. There are generally two ways to alleviate this issue: (1) increase the amount of training data and (2) add regularization to the model. For twitter and arxiv, we notice that the test F1 score doesn't seem to perform better past 8000 and 6000 training datapoints respectively (as shown by the vertical black lines). For these two cases, increasing the training data will not help close this gap, and thus the only way to reduce the gap is through regularization. The test F1 score for reddit keeps increasing though, so a further increase in training data may continue to close this gap. There is also the performance of the linear model to consider (Figure 5.13). Note that there is oscillation in the train metric for twitter and reddit. This is because the chosen model's hyper-parameters were fitted to the best performing validation score (not shown here) which can result in a decreased train score. There is still a gap between the apex of the train F1 score oscillation and the test F1 score. In each of the three sources, the test F1 score plateaus as outlined by the vertical black lines. Thus, a further increase in training data would not shrink this gap any further. The only way to shrink this gap would be to try further regularization of the model.

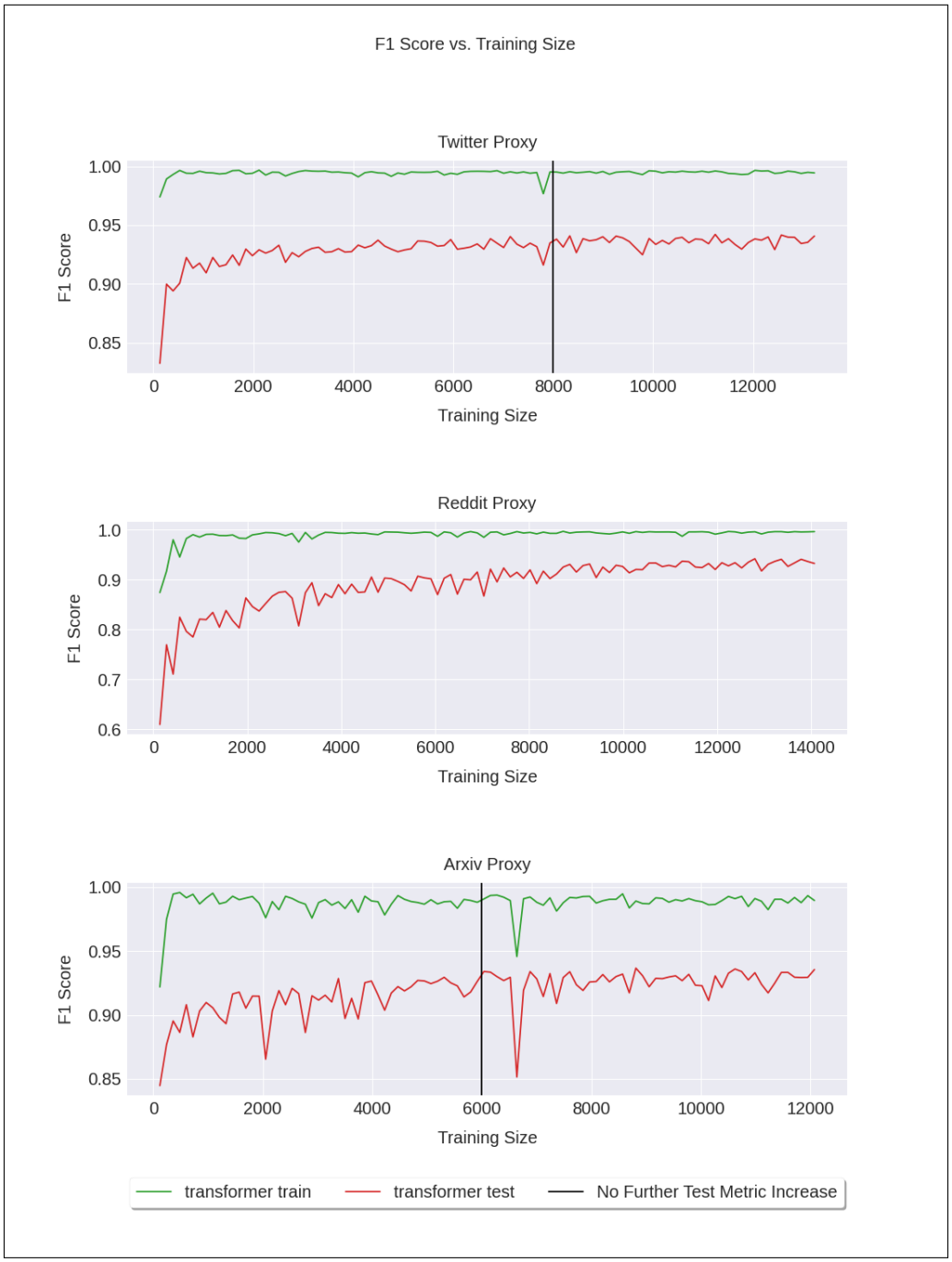


Figure 5.12: Increasing Train Size on Proxy Dataset - Transformer Model

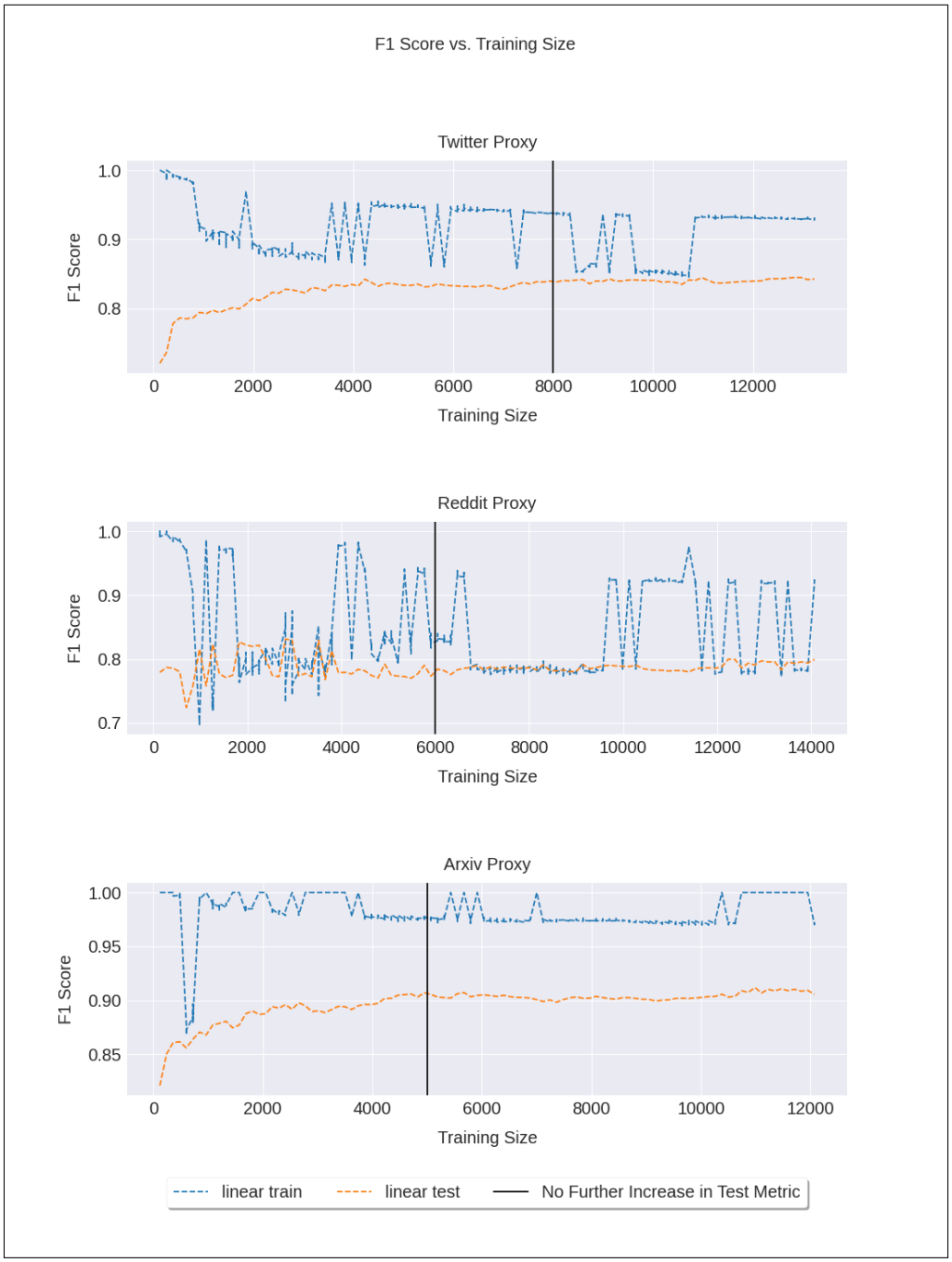


Figure 5.13: Increasing Train Size on Proxy Dataset - Linear Model

Research Questions Answered

Q. What is the Metric Performance difference between the Linear and Transformer Models?

A. With a training set of 2000+ training points, the transformer produces an average F1 test score (across the three sources in the proxy dataset) that is 7.7% higher than that of the linear model, while below 2000 data points, produces an average F1 test score that is 3.3% higher that of the linear model.

Q. What is the effect on Metric Performance as training size increases?

A. Increasing the training size, until approximately 6,000 data points, causes an increase in test metric performance. After this training size, the test metric performance plateaus.

5.4 Classification Threshold Effects on Metric Performance

This experiment aims to answer the research question "What is the effect of Classification Threshold on Metric Performance?" from Section 3.6.

We can see how the performance across metrics of F1 Score, Precision, and Recall are changed as we increase, or decrease, the classification threshold from 0.5 for both the linear model (Figure 5.14) and transformer model (Figure 5.15). Normally, an increase in threshold would increase precision and decrease recall (which may or may not alter the F1 Score).

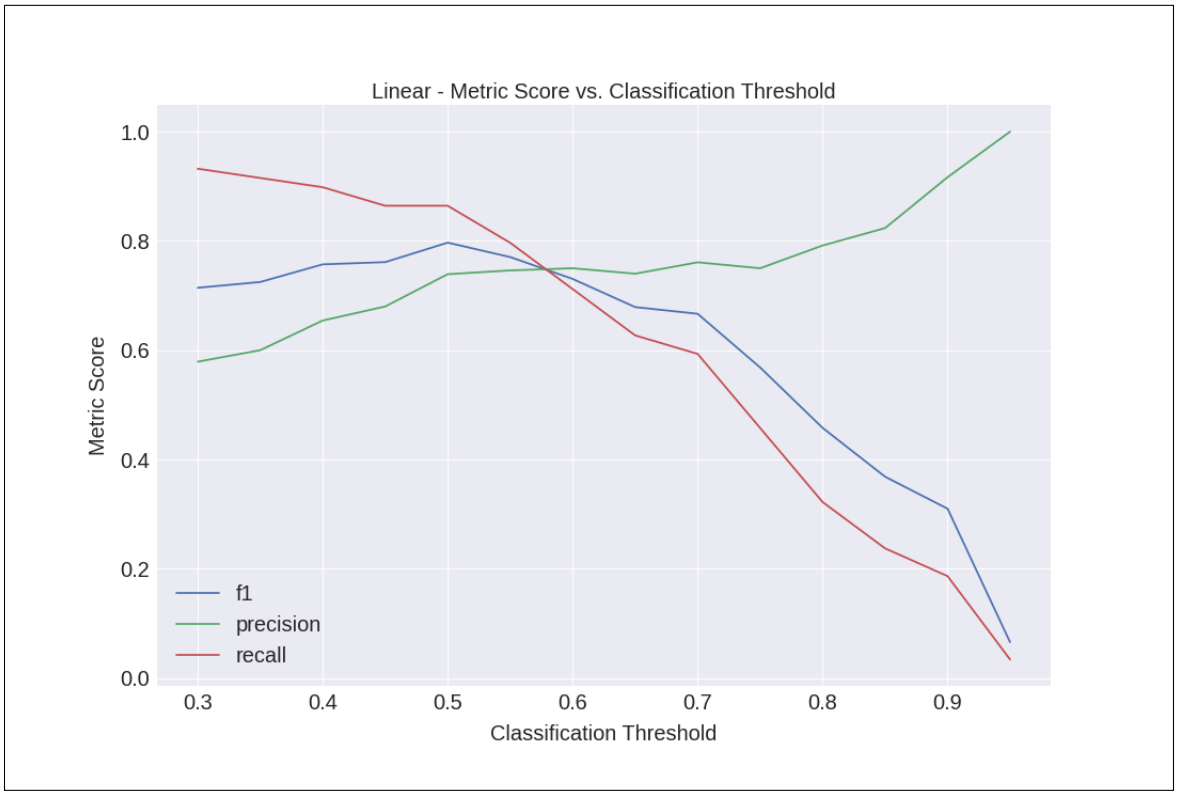


Figure 5.14: Metric vs. Classification Threshold - Linear Model



Figure 5.15: Metric vs. Classification Threshold - Transformer Model

In both cases, we can see that while increasing the classification threshold increases the precision, it causes a decrease in both the recall and F1 score. Thus, it does not seem wise to alter the classification threshold as the F1 score decreases.

Research Questions Answered

Q. What is the effect of Classification Threshold on Metric Performance?

A. Increasing, or decreasing, the threshold past 0.5 reduces the F1 Score.

5.5 Analysis of Model Predictions

This section aims to answer the research question of "Do the False Positive and False Negative predictions make sense?" from Section 3.7.

A visual quantitative analysis of term significance to model predictions can be found in the proceeding figures for both the linear and transformer models. Higher opacity denotes a larger magnitude, while a green or red color represent if the term contributes to a positive or negative prediction respectively.

Linear Model

False Positives

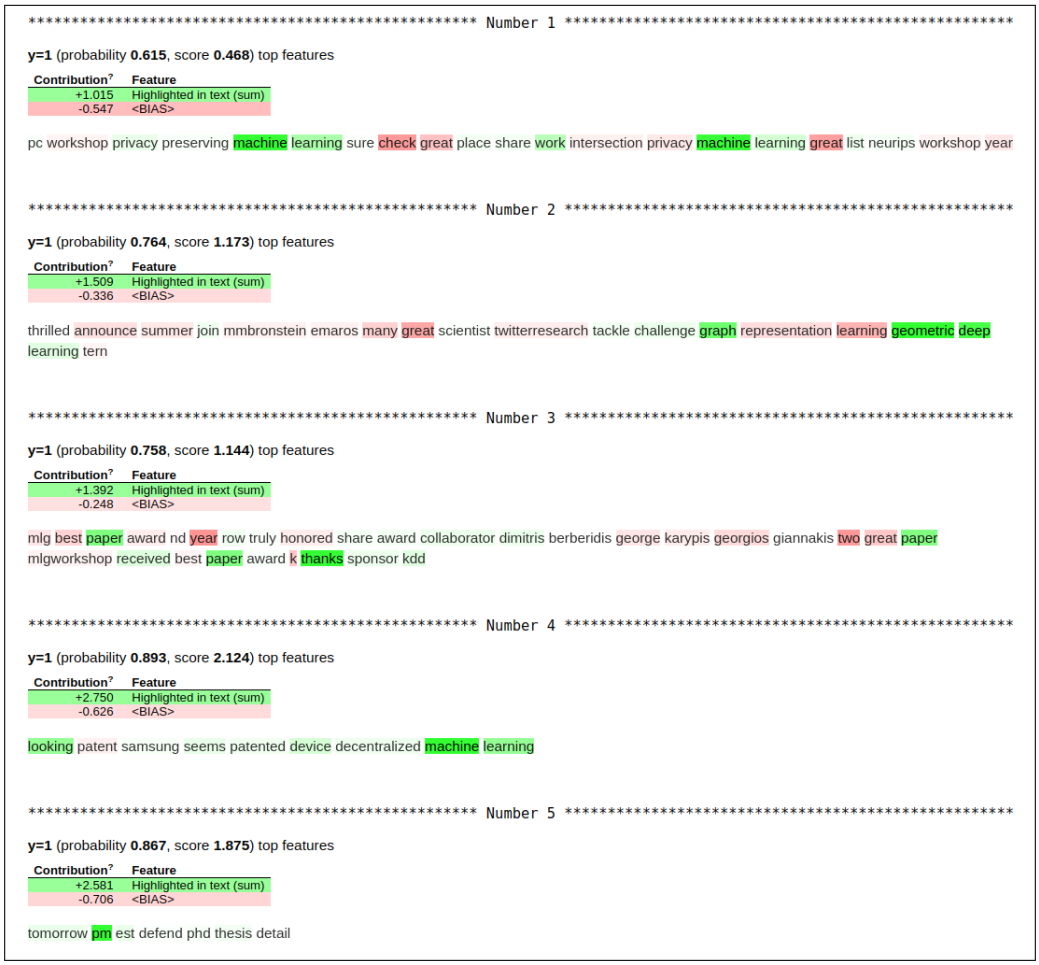


Figure 5.16: Linear Model False Positives 1 to 5



Figure 5.17: Linear Model False Positives 6 to 10



Figure 5.18: Linear Model False Positives 11 to 13

********** Number 14 **********

y=1 (probability **0.713**, score **0.910**) top features

Contribution[?]	Feature
+1.071	Highlighted in text (sum)
-0.160	<BIAS>

understanding spectral graph neural network graph neural network developed leap bound recent year due restriction traditional convolutional filter non euclidean structured data spectral graph theory mainly study fundamental graph property using algebraic method analyze spectrum adjacency matrix graph lay foundation graph convolutional neural network report note self contained come phd first year report literature review part illustrates link fundamental spectral graph theory graph convolutional neural network theory discus major spectral based graph convolutional neural network practical application graph neural network defined spectral domain also reviewed

********** Number 15 **********

y=1 (probability **0.640**, score **0.577**) top features

Contribution[?]	Feature
+0.738	Highlighted in text (sum)
-0.162	<BIAS>

relation spectral dimension ahlfors regular conformal dimension infinite graph spectral dimension weighted graph exponent associated asymptotic behavior random walk graph ahlfors regular conformal dimension dim mathrm ar graph distance quasisymmetric invariant quasisymmetry well studied property homeomorphisms metric space paper give counter example inequality dim mathrm ar le homogeneous scale symmetric class fractal also give sufficient condition inequality class scale symmetric graph

********** Number 16 **********

y=1 (probability **0.635**, score **0.554**) top features

Contribution[?]	Feature
+0.691	Highlighted in text (sum)
-0.137	<BIAS>

interpretable stability bound spectral graph filter graph structured data arise variety real world context ranging sensor transportation biological social network ubiquitous tool process graph structured data spectral graph filter used solve common task denoising anomaly detection well design deep learning architecture graph neural network despite important tool lack theoretical understanding stability property spectral graph filter important designing robust machine learning model paper study filter stability provide novel interpretable upper bound change filter output bound expressed term endpoint degree deleted newly added edge well spatial proximity edge upper bound allows u reason term structural property graph spectral graph filter stable perform extensive experiment verify intuition gained bound

Figure 5.19: Linear Model False Positives 14 to 16



Figure 5.20: Linear Model False Positives 17 to 18

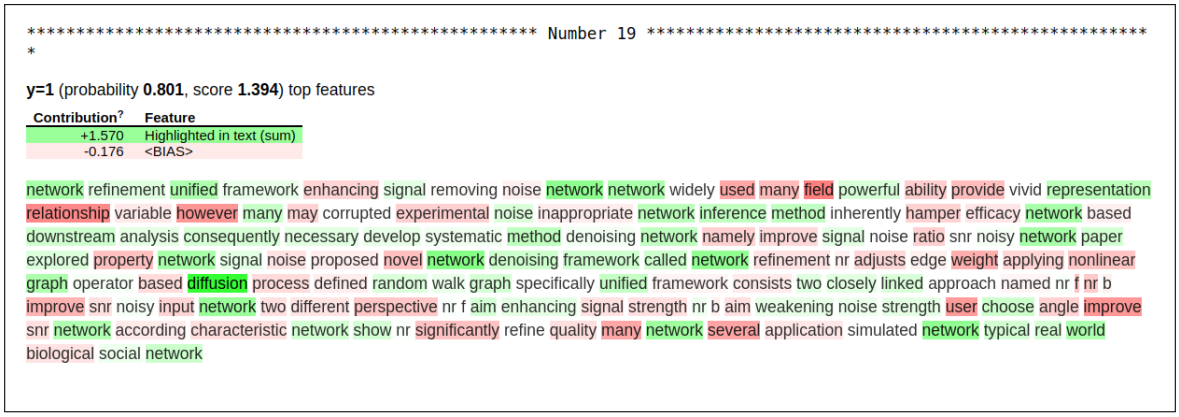


Figure 5.21: Linear Model False Positive 19

False Positive Number	**Analysis**	**Reasonable Decision?**
1	Gives positive weight to "machine learning" which makes sense.	✓
2	Gives positive weight to "graph", "geometric", and "deep" which make sense; although, gives negative weight to "representation learning" which doesn't make sense.	✓
3	Gives positive weight to "paper", which makes sense.	✓
4	Gives positive weight to "machine learning", which makes sense, but also the term "looking" which doesn't make much sense.	✓
5	Gives positive weight to "pm" which doesn't make much sense.	✗
6	Gives positive weight to "theory" which makes sense.	✓
7	Gives positive weight to "found" and "read" and negative weight to "researcher". None of these make sense.	✗
8	Gives larger positive weight to the terms "paper", "neural", and "deep" which makes sense.	✓
9	Gives larger positive weight to the terms "local", "graph", and "clustering" which makes sense.	✓
10	Gives positive weight to the terms "paper", a reasonable choice, and "order", which is not as intuitive.	✓
11	Gives positive weight to the terms "deep", "network", and "machine learning" which makes sense.	✓
12	Gives positive weight to "machine learning", which makes sense, and "really", which doesn't make much sense.	✓
13	Gives a positive weight to "graph", "node" and "diffusion" which makes sense.	✓

14	Gives positive weight to "graph" and "network" which makes sense.	✓
15	Gives positive weight to "graph" and "invariant" which makes sense.	✓
16	Gives positive weight to "graph", "deep", "network" and "machine" which makes sense.	✓
17	Gives positive weight to "network", "node" and "deep" which makes sense.	✓
18	Gives positive weight to "graph" and "gnns" which makes sense.	✓
19	Gives positive weight to "network" and "diffusion" which makes sense.	✓

Table 5.1: Analysis of Linear Model False Positives

False Negatives



Figure 5.22: Linear Model False Negatives 1 to 4



Figure 5.23: Linear Model False Negatives 5 to 8

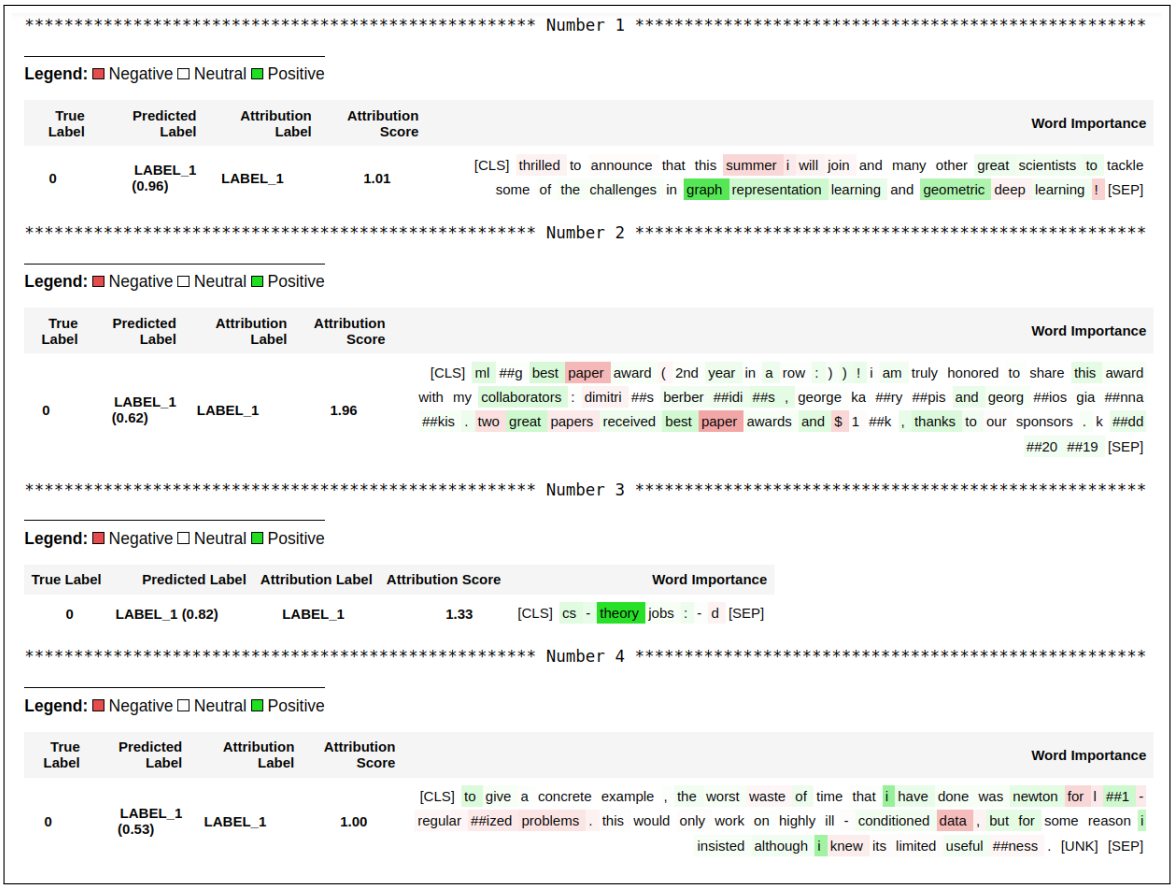
False Negative Number	Analysis	Reasonable Decision?
1	Gives negative weight to "free" and "application" which is reasonable.	✓
2	Gives negative weight to "group" which doesn't make much sense. This should be a neutral word.	✗
3	Gives negative weight to "event" and "unsupervised learning" which should be neutral and positive respectively.	✗

4	This is missing some link information (which doesn't show up in tweet text) so this makes sense it was a false negative.	✓
5	Gives negative weight to "possible", "interest", and "processing". These should likely be neutral words instead of negative.	✗
6	Gives negative weight to "conference" and "data" which doesn't make sense.	✗
7	Gives negative weight to "process" which should probably just be neutral.	✗
8	Gives negative weight to "neural", which doesn't seem right, and "et al" which makes sense.	✓

Table 5.2: Analysis of Linear Model False Negatives

Transformer Model

False Positives



********** Number 1 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.96)	LABEL_1	1.01	[CLS] thrilled to announce that this summer i will join and many other great scientists to tackle some of the challenges in graph representation learning and geometric deep learning ! [SEP]

********** Number 2 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.62)	LABEL_1	1.96	[CLS] ml ##g best paper award (2nd year in a row :)) ! i am truly honored to share this award with my collaborators : dimitri ##s berber ##idi ##s , george ka ##ry ##pis and georg ##ios gia ##nna ##kis . two great papers received best paper awards and $ 1 ##k , thanks to our sponsors . k ##dd ##20 ##19 [SEP]

********** Number 3 **********

Legend: Negative Neutral Positive

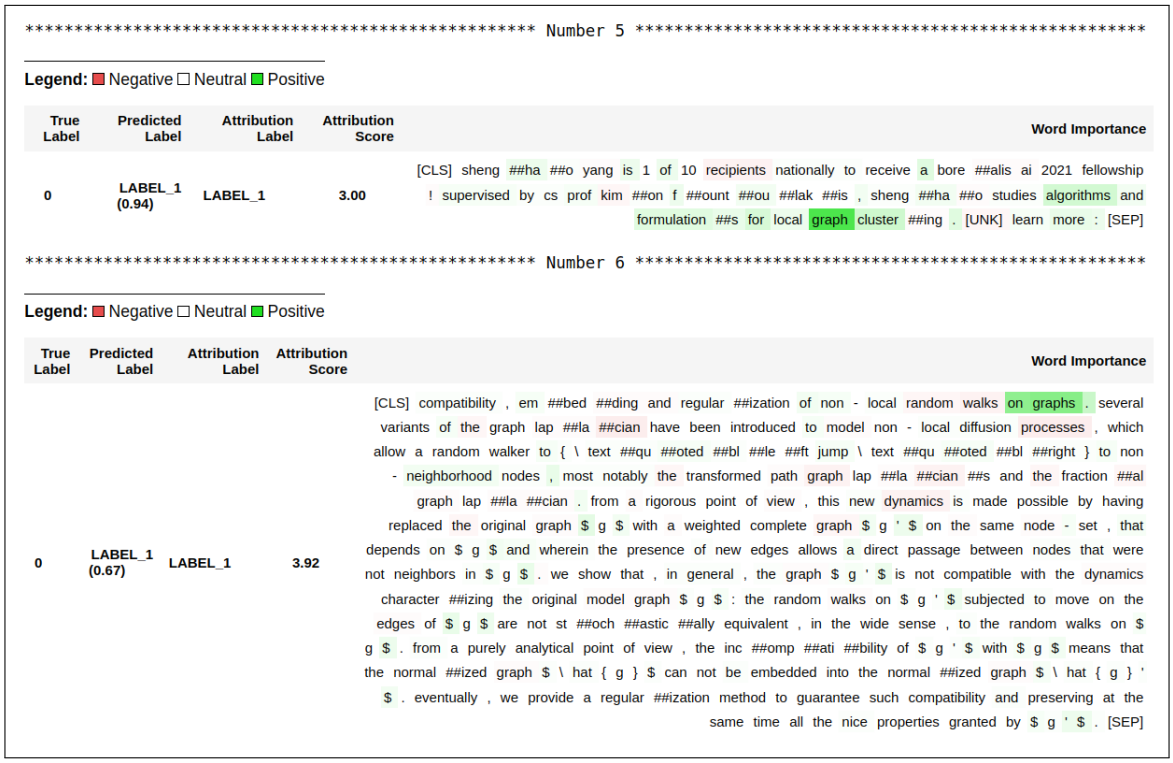
True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.82)	LABEL_1	1.33	[CLS] cs - theory jobs : - d [SEP]

********** Number 4 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.53)	LABEL_1	1.00	[CLS] to give a concrete example , the worst waste of time that i have done was newton for l ##1 - regular ##ized problems . this would only work on highly ill - conditioned data , but for some reason i insisted although i knew its limited useful ##ness . [UNK] [SEP]

Figure 5.24: Transformer Model False Positives 1 to 4



********** Number 5 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.94)	LABEL_1	3.00	[CLS] sheng ##ha ##o yang is 1 of 10 recipients nationally to receive a bore ##alis ai 2021 fellowship ! supervised by cs prof kim ##on f ##ount ##ou ##lak ##is , sheng ##ha ##o studies algorithms and formulation ##s for local graph cluster ##ing . [UNK] learn more : [SEP]

********** Number 6 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.67)	LABEL_1	3.92	[CLS] compatibility , em ##bed ##ding and regular ##ization of non - local random walks on graphs . several variants of the graph lap ##la ##cian have been introduced to model non - local diffusion processes , which allow a random walker to { \ text ##qu ##oted ##bl ##le ##ft jump \ text ##qu ##oted ##bl ##right } to non - neighborhood nodes , most notably the transformed path graph lap ##la ##cian ##s and the fraction ##al graph lap ##la ##cian . from a rigorous point of view , this new dynamics is made possible by having replaced the original graph $ g $ with a weighted complete graph $ g ' $ on the same node - set , that depends on $ g $ and wherein the presence of new edges allows a direct passage between nodes that were not neighbors in $ g $. we show that , in general , the graph $ g ' $ is not compatible with the dynamics character ##izing the original model graph $ g $: the random walks on $ g ' $ subjected to move on the edges of $ g $ are not st ##och ##astic ##ally equivalent , in the wide sense , to the random walks on $ g $. from a purely analytical point of view , the inc ##omp ##ati ##bility of $ g ' $ with $ g $ means that the normal ##ized graph $ \ hat { g } $ can not be embedded into the normal ##ized graph $ \ hat { g } ' $. eventually , we provide a regular ##ization method to guarantee such compatibility and preserving at the same time all the nice properties granted by $ g ' $. [SEP]

Figure 5.25: Transformer Model False Positives 5 to 6

********** Number 7 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.83)	LABEL_1	2.19	[CLS] understanding spectral graph neural network . the graph neural networks have developed by leaps and bounds in recent years due to the restriction of traditional con ##vo ##lu ##tion ##al filters on non - euclidean structured data . spectral graph theory mainly studies fundamental graph properties using algebraic methods to analyze the spectrum of the ad ##ja ##cen ##cy matrix of a graph , which lays the foundation of graph con ##vo ##lu ##tion ##al neural networks . this report is more than notes and self - contained which comes from my ph . d . first - year report literature review part , it illustrates how to link fundamental ##s of spectral graph theory to graph con ##vo ##lu ##tion ##al neural network theory , and discusses the major spectral - based graph con ##vo ##lu ##tion ##al neural networks . the practical applications of the graph neural networks defined in the spectral domain is also reviewed . [SEP]

********** Number 8 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.66)	LABEL_1	2.59	[CLS] constant curvature graph con ##vo ##lu ##tion ##al networks . interest has been rising lately towards methods representing data in non - euclidean spaces , e . g . hyper ##bolic or spherical that provide specific ind ##uc ##tive bias ##es useful for certain real - world data properties , e . g . scale - free , hierarchical or cyclic ##al . however , the popular graph neural networks are currently limited in modeling data only via euclidean geometry and associated vector space operations . here , we bridge this gap by proposing mathematical ##ly grounded general ##izations of graph con ##vo ##lu ##tion ##al networks (g ##c ##n) to (products of) constant curvature spaces . we do this by i) introducing a unified formal ##ism permitting a different ##iable inter ##pol ##ation between all geo ##met ##ries of constant curvature ir ##res ##pe ##ctive of their sign , ii) lever ##aging g ##yr ##o - bar ##yce ##nt ##ric coordinates that general ##ize the classic euclidean concept of the center of mass . our class of models smoothly recover their euclidean counterparts when the curvature goes to zero from either side . empirical ##ly , we out ##per ##form euclidean g ##c ##ns in the tasks of node classification and distortion mini ##mi ##zation for symbolic data exhibiting non - euclidean behavior , according to their discrete curvature . [SEP]

Figure 5.26: Transformer Model False Positives 7 to 8

********** Number 9 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.62)	LABEL_1	2.76	[CLS] the eu ##ler characteristic : a general topological des ##cript ##or for complex data . data ##set ##s are mathematical objects (e . g . , point clouds , matrices , graphs , images , fields / functions) that have shape . this shape en ##codes important knowledge about the system under study . topology is an area of mathematics that provides diverse tools to character ##ize the shape of data objects . in this work , we study a specific tool known as the eu ##ler characteristic (ec) . the ec is a general , low - dimensional , and interpret ##able des ##cript ##or of topological spaces defined by data objects . we rev ##ise the mathematical foundations of the ec and highlight its connections with statistics , linear algebra , field theory , and graph theory . we discuss advantages offered by the use of the ec in the characterization of complex data ##set ##s ; to do so , we illustrate its use in different applications of interest in chemical engineering such as process monitoring , flow cy ##tom ##et ##ry , and microscopy . we show that the ec provides a des ##cript ##or that effectively reduces complex data ##set ##s and that this reduction facilitates tasks such as visual ##ization , regression , classification , and cluster ##ing . [SEP]

********** Number 10 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
0	LABEL_1 (0.97)	LABEL_1	7.37	[CLS] on explain ##ability of graph neural networks via sub ##graph exploration ##s . we consider the problem of explaining the predictions of graph neural networks (g ##nn ##s) , which otherwise are considered as black boxes . existing methods invariably focus on explaining the importance of graph nodes or edges but ignore the sub ##st ##ru ##cture ##s of graphs , which are more intuitive and human - intel ##li ##gible . in this work , we propose a novel method , known as sub ##graph ##x , to explain g ##nn ##s by identifying important sub ##graphs . given a trained g ##nn model and an input graph , our sub ##graph ##x explains its predictions by efficiently exploring different sub ##graphs with monte carlo tree search . to make the tree search more effective , we propose to use sha ##ple ##y values as a measure of sub ##graph importance , which can also capture the interactions among different sub ##graphs . to ex ##ped ##ite computation ##s , we propose efficient approximation schemes to compute sha ##ple ##y values for graph data . our work represents the first attempt to explain g ##nn ##s via identifying sub ##graphs explicitly and directly . experimental results show that our sub ##graph ##x achieve ##s significantly improved explanations , while keeping computation ##s at a reasonable level . [SEP]

Figure 5.27: Transformer Model False Positives 9 to 10

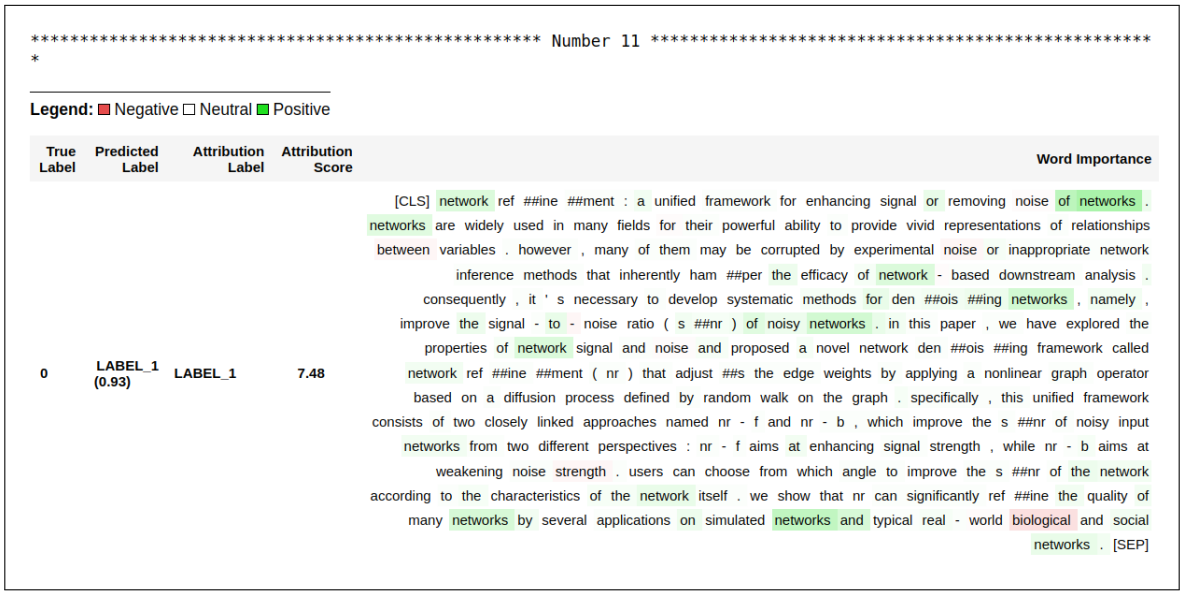


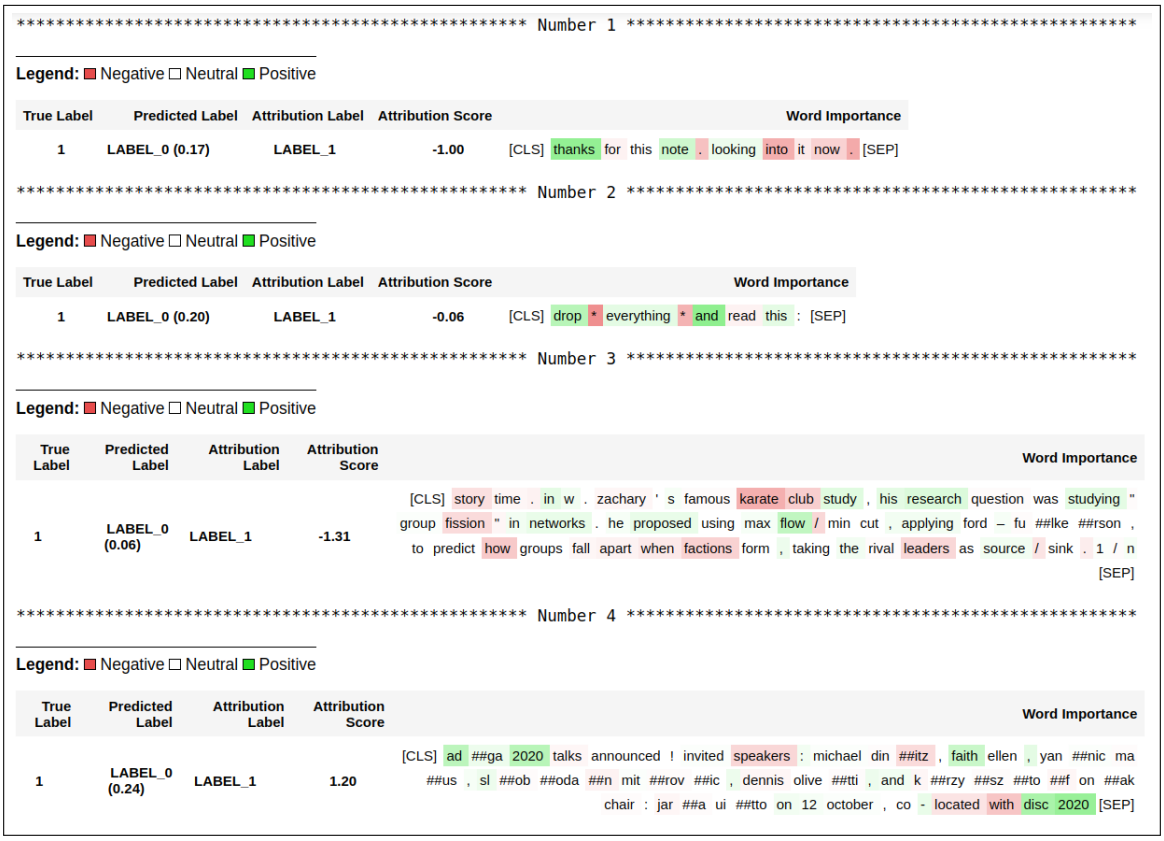
Figure 5.28: Transformer Model False Positive 11

False Positive Number	**Analysis**	**Reasonable Decision?**
1	Gives positive weight to ”graph”, ”representation”, ”learning”, and ”geometric” which makes sense.	✓
2	Gives positive weight to ”best”, ”thanks”, and ”great” which should be neutral.	✗
3	Gives positive weight to ”theory” which makes sense.	✓
4	Gives positive weight to ”i” which doesn’t make much sense.	✗
5	Gives positive weight to ”graph”, ”cluster” and ”algorithms” which makes sense.	✓
6	Gives positive weight to ”graph” which makes sense.	✓

7	Gives positive weight to "spectral" which is reasonable, but for some reason gives negative weight to "graph".	✗
8	Gives positive weight to "network" which makes sense but negative weight to "graph" which doesn't.	✗
9	Gives positive weight to "chemical" which doesn't make much sense.	✗
10	Gives positive weight to "graph" which makes sense.	✓
11	Gives positive weight to "networks" which makes sense.	✓

Table 5.3: Analysis of Transformer Model False Positives

False Negatives



********** Number 1 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.17)	LABEL_1	-1.00	[CLS] thanks for this note . looking into it now . [SEP]

********** Number 2 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.20)	LABEL_1	-0.06	[CLS] drop * everything * and read this : [SEP]

********** Number 3 **********

Legend: Negative Neutral Positive

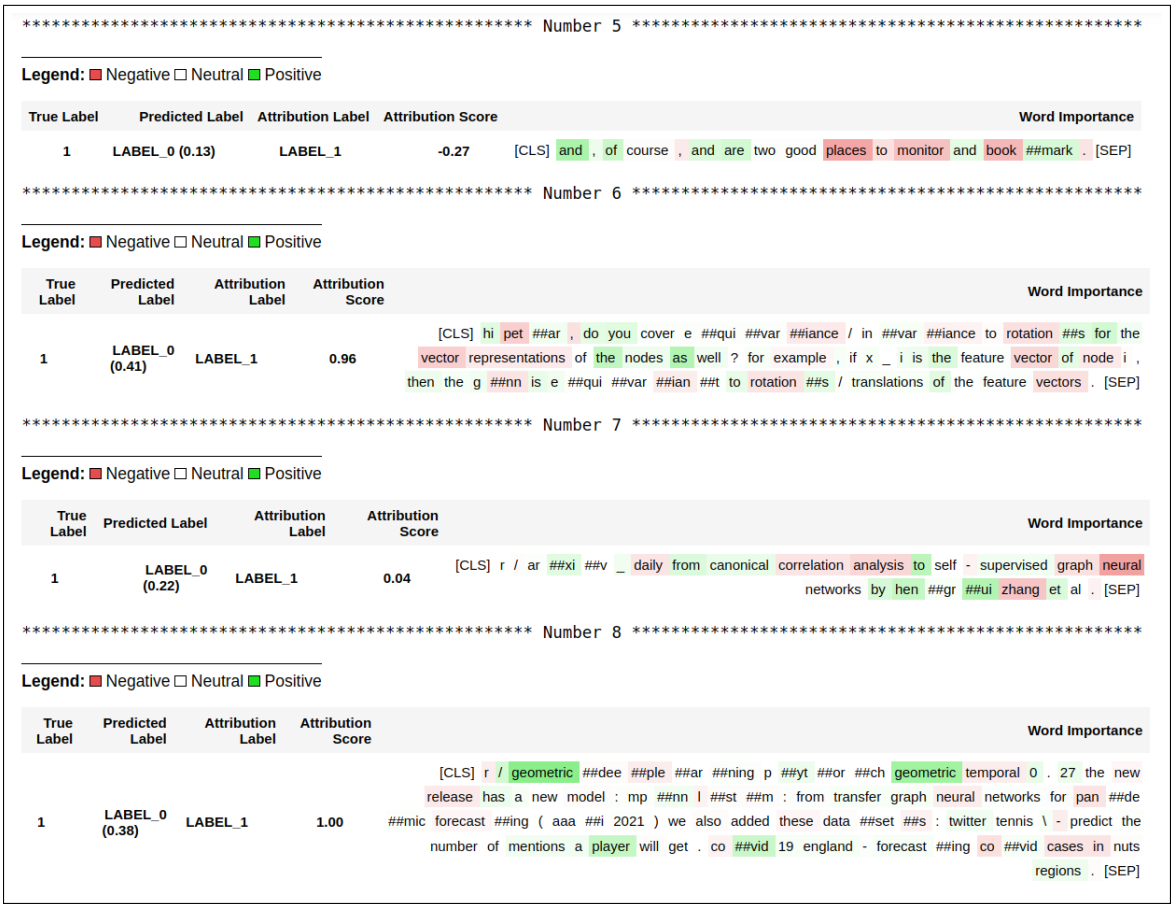
True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.06)	LABEL_1	-1.31	[CLS] story time . in w . zachary ' s famous karate club study , his research question was studying " group fission " in networks . he proposed using max flow / min cut , applying ford – fu ##lke ##rson , to predict how groups fall apart when factions form , taking the rival leaders as source / sink . 1 / n [SEP]

********** Number 4 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.24)	LABEL_1	1.20	[CLS] ad ##ga 2020 talks announced ! invited speakers : michael din ##itz , faith ellen , yan ##nic ma ##us , sl ##ob ##oda ##n mit ##rov ##ic , dennis olive ##tti , and k ##rzy ##sz ##to ##f on ##ak chair : jar ##a ui ##tto on 12 october , co - located with disc 2020 [SEP]

Figure 5.29: Transformer Model False Negatives 1 to 4



********** Number 5 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.13)	LABEL_1	-0.27	[CLS] and , of course , and are two good places to monitor and book ##mark . [SEP]

********** Number 6 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.41)	LABEL_1	0.96	[CLS] hi pet ##ar , do you cover e ##qui ##var ##iance / in ##var ##iance to rotation ##s for the vector representations of the nodes as well ? for example , if x _ i is the feature vector of node i , then the g ##nn is e ##qui ##var ##ian ##t to rotation ##s / translations of the feature vectors . [SEP]

********** Number 7 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.22)	LABEL_1	0.04	[CLS] r / ar ##xi ##v _ daily from canonical correlation analysis to self - supervised graph neural networks by hen ##gr ##ui zhang et al . [SEP]

********** Number 8 **********

Legend: Negative Neutral Positive

True Label	Predicted Label	Attribution Label	Attribution Score	Word Importance
1	LABEL_0 (0.38)	LABEL_1	1.00	[CLS] r / geometric ##dee ##ple ##ar ##ning p ##yt ##or ##ch geometric temporal 0 . 27 the new release has a new model : mp ##nn l ##st ##m : from transfer graph neural networks for pan ##de ##mic forecast ##ing (aaa ##i 2021) we also added these data ##set ##s : twitter tennis \ - predict the number of mentions a player will get . co ##vid 19 england - forecast ##ing co ##vid cases in nuts regions . [SEP]

Figure 5.30: Transformer Model False Negatives 5 to 8

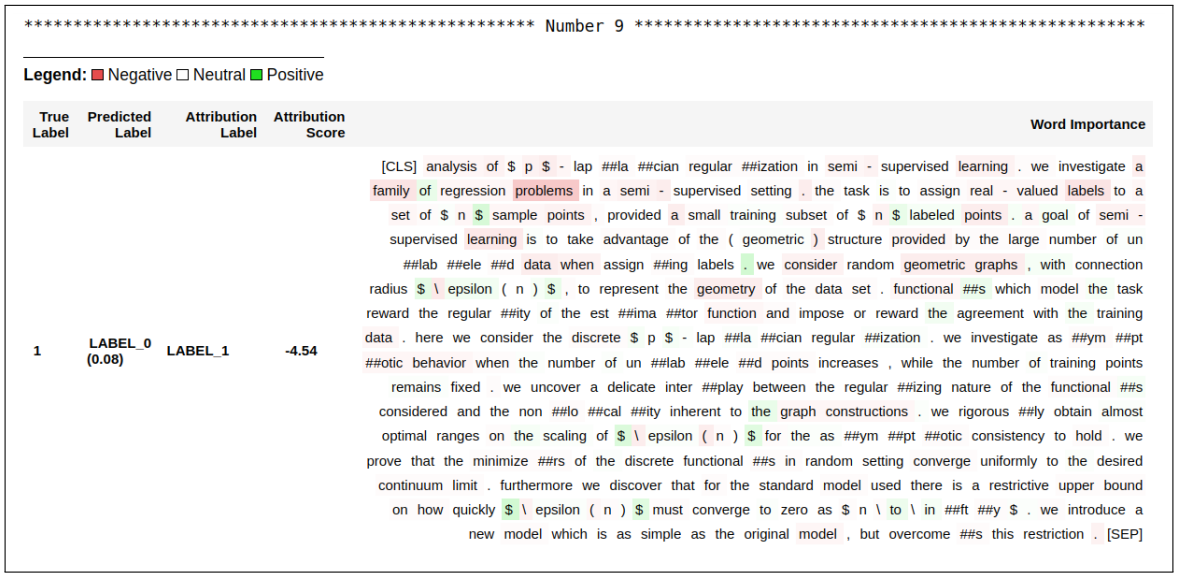


Figure 5.31: Transformer Model False Negatives 9

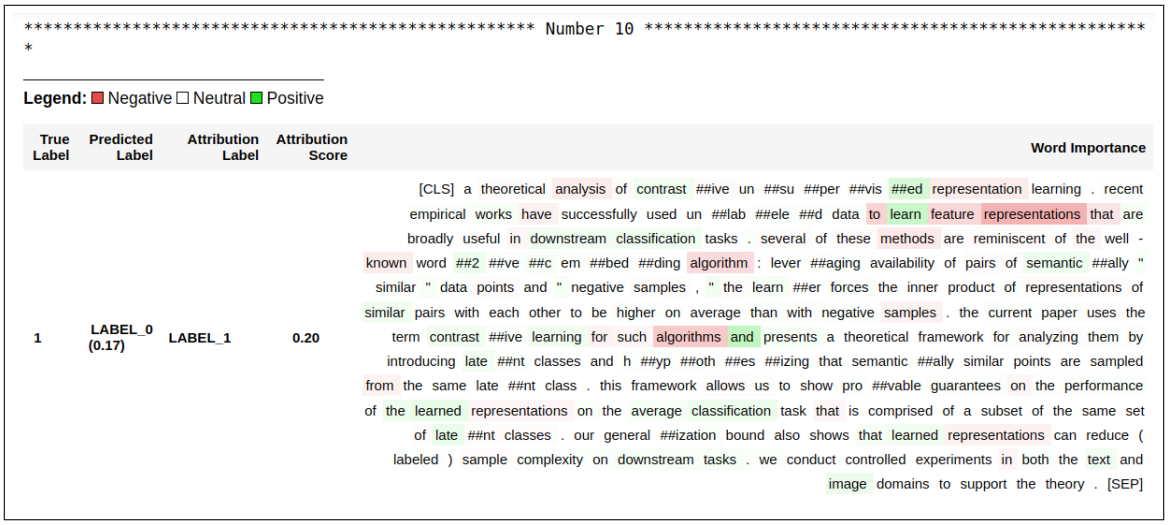


Figure 5.32: Transformer Model False Negatives 10

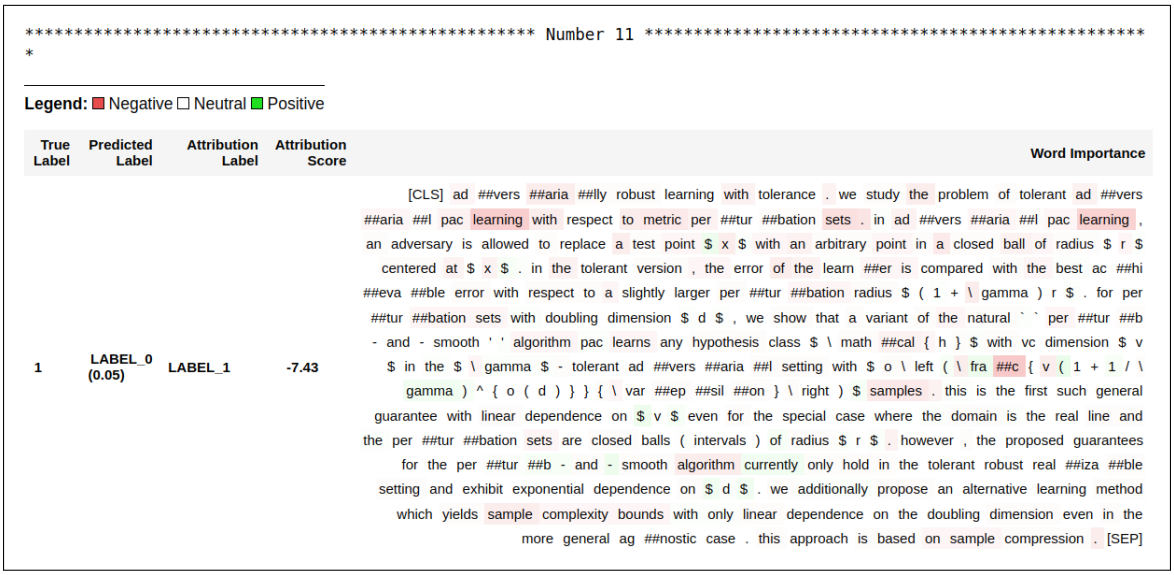


Figure 5.33: Transformer Model False Negative 11

False Negative Number	Analysis	Reasonable Decision?
1	Based on subject matter, a negative label makes sense here. It seems this is a reply to a tweet that is missing some important information.	✓
2	Link is missing from text so this makes sense it would be labelled as negative.	✓
3	Gives negative weight to terms such as "story", "time", "faction", "karate", and "club". Overall this seems like a reasonable interpretation.	✓
4	Both the positively weighted terms, such as "ad", "2020" and "faith", and negatively weighted terms, such as "speakers", "located" and "with", don't really make a lot of sense. It seems they should be more neutral.	✗
5	Important link information isn't included in the text; therefore, this interpretation makes sense.	✓

6	Gives negative weights to terms "vector" and "representations", while giving positive weights to terms "the" and "as". This doesn't make much sense.	✗
7	Gives negative weights to "graph" and "neural" which doesn't make sense.	✗
8	Gives positive weights to "geometric", which makes sense, and negative weights to "neural", which doesn't make sense.	✗
9	Gives negative weights to "geometric", "graphs", "learning" and "data" which does not make sense. Also gives positive weight to "$" which should be neutral.	✗
10	Gives positive weights to "downstream", "learning" and "classification", which makes sense, but negative weights to "feature", "representations" and "algorithms" which doesn't make sense.	✗
11	Gives negative weights to "learning", "algorithm", and "sample" which should be neutral or slightly positive.	✗

Table 5.4: Analysis of Transformer Model False Negatives

Research Questions Answered

Q. Do the False Positive and False Negative predictions appear logical?

A. Many of the False Positives and False Negatives of both models appear to be logical, but many also seem illogical based on some keywords found in the text. Therefore, further investigation into these discrepancies must be looked into.

5.6 Temporal Analysis of Sample Dataset Performance

This experiment aims to answer the research question "What is the performance on future recommendations based on past labels?" (Section 3.8).

In order to test this, all rows in the sample dataset were sorted by the date in which they were labeled. Then a train test split similar to Section 3.3 was performed. The test F1 scores for this experiment (called temporal) and that of Section 3.3 (called non-temporal) for both the linear and transformer model can be found in Figures 5.34 and 5.35 respectively.

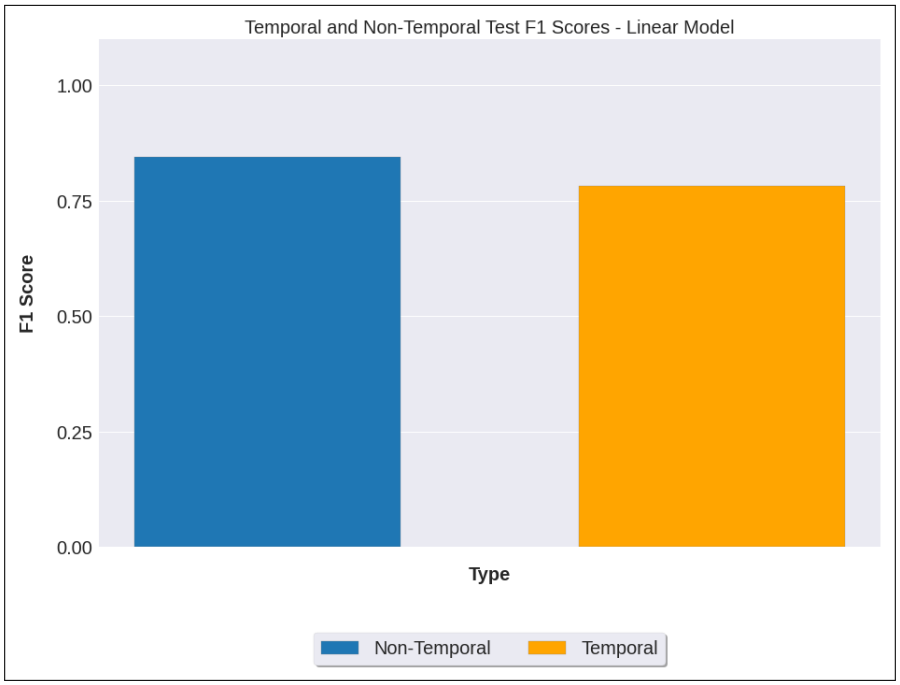


Figure 5.34: Temporal and Non-Temporal Test F1 Scores - Linear Model

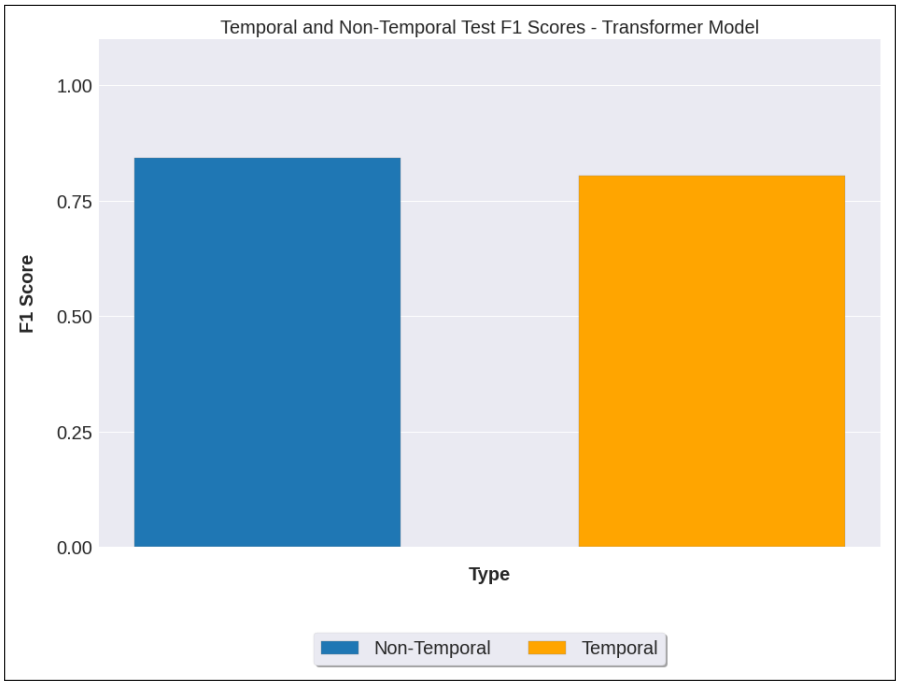


Figure 5.35: Temporal and Non-Temporal Test F1 Scores - Transformer Model

We can see for both models, the temporal experiment has a worse performance. While it is possible that the results from Section 3.3 did have look ahead bias, it is also possible the sample dataset is just susceptible to noise because of its relatively small size.

Research Questions Answered

Q. What is the performance on future recommendations based on past labels?

A. The temporal experiment did not perform as well as the non-temporal experiment; however, this is possibly just due to noise from the sample dataset size and not an indication of look ahead bias. Future experiments would need to be performed as more user data is acquired.

Chapter 6

Conclusion

Up2Date Research's goal is to help academic researchers stay current in their field of research by both reducing the amount of time spent researching across and within sources by aggregating and filtering this data with a personalized machine learning model. In contrast to the related works of Arxiv Sanity, Google Scholar, Research Rabbit and Deeplearn, Up2Date Research provides users with recommendations from multiple sources with the ability to improve upon these recommendations via user feedback.

Model performance on the sample dataset achieved an F1 Score of 82% which is promising for the value proposition of this application to its users. Further improvement is always possible and future work is detailed in Appendix D.

References

[1] C. G. Michael Fire, "Over-optimization of academic publishing metrics: Observing goodhart's law in action," *Giga Science*, vol. 8, no. 6, pp. 1–20, 2019.

[2] Kaggle, "Tweet sentiment analysis," 2020, url=https://www.kaggle.com/competitions/tweet-sentiment-extraction/data.

[3] C. Kumar, "Twitter and reddit sentimental analysis dataset," 2019, url=https://www.kaggle.com/datasets/cosmos98/twitter-and-reddit-sentimental-analysis-dataset.

[4] A. Karpathy, "arxiv-sanity," 2023, url=https://arxiv-sanity-lite.com/.

[5] Google, "Google scholar," 2023, url=https://scholar.google.ca/.

[6] R. Rabbit, "Research rabbit," 2023, url=https://researchrabbit.ai.

[7] R. Shu, "Deep learning monitor," 2017, url=https://deeplearn.org/.

[8] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2019. [Online]. Available: https://arxiv.org/abs/1910.01108

[9] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018. [Online]. Available: https://arxiv.org/abs/1810.04805

[10] T. Chen and C. Guestrin, "XGBoost," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, aug 2016. [Online]. Available: https://doi.org/10.1145%2F2939672.2939785

[11] R. T. Jerome Friedman, Trevor Hastie, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, pp. 1–22, 2010.

[12] Scikit-Learn, "Support vector machines - scikit-learn," 2022, url=https://scikit-learn.org/stable/modules/svm.html#classification.

[13] ——, "Naive bayes," 2022, url=https://scikit-learn.org/stable/modules/naive_bayes.html.

[14] ——, "Text feature extraction," 2022, url=https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction.

[15] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[16] S. N. Group, "Stanford tokenizer," 2022, url=https://nlp.stanford.edu/software/tokenizer.shtml.

[17] D. Cer, Y. Yang, S. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar, Y. Sung, B. Strope, and R. Kurzweil, "Universal sentence encoder," *CoRR*, vol. abs/1803.11175, 2018. [Online]. Available: http://arxiv.org/abs/1803.11175

[18] Sklearn, "Metrics and scoring: quantifying the quality of predictions," 2022, url=https://scikit-learn.org/stable/modules/model_evaluation.html.

[19] M. AI, "Text classification," 2023, url=https://paperswithcode.com/task/text-classification.

[20] D. E. Knuth, "Literate programming," *The Computer Journal*, vol. 27, no. 2, pp. 97–111, 1984.

[21] M. P. Inc., "Reactjs - a javascript library for building user interfaces," 2022, url=https://reactjs.org/.

[22] M. Erikson, "Idiomatic redux: The history and implementation of react-redux," 2018, url=https://blog.isquaredsoftware.com/2018/11/react-redux-history-implementation/.

[23] M. U. SAS, "Mui: The react component library you always wanted," 2022, url=https://mui.com/.

[24] Flask, "Welcome to flask - flask documentation 2.2.x," 2010, url=https://flask.palletsprojects.com/en/2.2.x/.

[25] RedHat, "What is the rest api?" 2020, url=https://www.redhat.com/en/topics/api/what-is-a-rest-api.

[26] R. Ltd., "What is redis?" 2022, url=https://redis.io/docs/about/.

[27] M. Bayer, "Sqlalchemy - the database toolkit for python," 2022, url=https://www.sqlalchemy.org/.

[28] D. Inc., "Docker: Accelerated, containerized application development," 2022, url=https://www.docker.com/.

[29] J. Roesslein, "Tweepy documentation," 2022, url=https://docs.tweepy.org/en/stable/.

APPENDICES

Appendix A

Infrastructure

This chapter discusses the infrastructure of the application, including the backend, client, database and remote server.

A.1 Client

The client of the application was written using React. React was chosen as it is able to create one-page web applications [21] that easily react to user inputs. This was vital as we are displaying a multitude of cross-platform posts in which the user will react with to provide labels.

A.1.1 Design

The react application was written using both functions and classes. The method of passing variables between the different components was primarily done by passing props from parents down to children. React Redux [22] was not leveraged at this time. In order to create a streamlined user interface, substantial use of the material UI [23] package was leveraged to create buttons, loading bars, date selections and more.

A.1.2 Leveraging Local Storage

In order to reduce the number of redundant REST API calls to the backend and the database, local storage of the web browser was leveraged. Local storage was used to store

the data for the following pages:

- Recommendations
- Likes
- Papers
- Keywords
- Settings

Local Storage Implementation

When a user first navigates to the recommendations, likes, papers, settings, or keywords page, the client will invoke a GET request to the backend which will always return the same data, unless the user alters it.

Therefore, when first loading any of these pages, the client first checks if there is a key value pair stored in local storage. This is accomplished with the help of custom JavaScript functions called ‘getLocalStorage‘ (Listing A.1) and ‘save2storage‘ (Listing A.2). ‘getLocalStorage‘ is first used to check whether a key value pair is stored in local storage.

```
 function getLocalStorage(item) {
/*
Item in local storage is saved as an object

{
    "data": ...,
    "timestamp": ...
}

If there item is saved and the timestamp is less than 10 minutes old,
then we return the data

Otherwise we remove the item from local storage (since it is stale)
and return null

*/

let item_string = localStorage.getItem(item);

if (item_string == null) {
```

```
        return null;
    }

    let item_object = JSON.parse(item_string);

    let timestamp = item_object["timestamp"];

    let elapsed_minutes = milliseconds2minutes(Date.now() - timestamp);

    let time_limit = GetEnvironmentVar(
        "REACT_APP_LOCAL_STORAGE_TIMEOUT_MINUTES",
        10
    );

    if (elapsed_minutes > time_limit) {
        localStorage.removeItem(item);
        return null;
    }

    console.log(item_object["data"]);
    // Otherwise Return the Data
    return item_object["data"];

}
```

Listing A.1: getLocalStorage Function

The function first checks if the key is in local storage. If not, it returns null. Then, if the key is in local storage it checks if the timestamp at which it was originally added is expired (an expiration of 10 minutes is used). If it is expired, then the item is removed from local storage and then the function returns null. Otherwise the value is returned.

If the function ‘getLocalStorage‘ returns null, then a REST api call is made to the backend to get the data. Once received, the item is saved to local storage using the function ‘save2storage‘.

```
     function save2storage(name, value) {

    let x = {
        data: value,
        timestamp: Date.now()
    }

    x = JSON.stringify(x);

    localStorage.setItem(
```

```
        name,
        x
    )

    return;
}
```

Listing A.2: save2storage Function

'save2storage' takes the key and value pair for the item to be stored in local storage. It also adds a timestamp to the value so that the function 'getLocalStorage' can use it afterward.

Local Storage Removal

Local storage will be removed if the user alters the data. For recommendations and likes, the local storage will be removed if the user labels or dislikes, respectively, any posts as this post will no longer be shown on reload and another post will take its place. For papers, local storage will be removed if the user submits new paper urls, or if they dislike a paper. For keywords and settings, the local storage will be removed if the user presses the save button. Finally, all items will be removed from local storage if the user logs out of the application.

A.2 Backend

The backend for this application was written using Flask [24]. Flask was chosen as it is a popular micro web framework written in Python.

A.2.1 REST API

The backend handles all server-side computation using the REST API [25]. There are two main categories of RESTful resources that were made: (1) authentication resource and (2) data traversal resource. Authentication calls include sign-up, login, logout and session confirmation. Data traversal includes receiving post recommendations, viewing liked posts, adding liked papers, viewing liked papers, sending labels, altering settings, altering keywords and handling pagination.

Authentication calls are handled under the '/auth/' URI (Uniform Resource Identifier) while data traversal calls are handled under the '/api/' URI.

AUTH Resource

This section will detail all of the authorization calls made using the REST api.

Sign-Up

To sign-up for the application, the client would make a POST request to the '/auth/sign-up' URI. Assuming there were no issues on the backend, the user will then be invited to login with the newly created credentials.

Sending Request

An example of the body for the POST request can be found in Listing A.3.

```
{
    "email": "chris@hotmail.com",
    "password1": "password",
    "password2": "password"
}
```

Listing A.3: Sign-Up POST Request Payload

Return of Request

The return request will be in the form of a JSON. If the user already exists (same email) then the response in Listing A.4 will be received.

```
Response(
    response=json.dumps({
        "reason": "User already exists"
    }),
    status=401,
    mimetype="application/json",

)
```

Listing A.4: Sign-Up POST Request Failure Response

Otherwise, the user will be added to the database and the response in Listing A.5 will be sent back.

```
Response(
    response=json.dumps({
        "status": 200
    }),
    status=200,
    mimetype="application/json"
)
```

Listing A.5: Sign-Up POST Request Success Response

Database Changes

Assuming the information the user entered in the sign-up page meets all specifications (such as password length, proper email, etc.) then a hash map is created and the user is added to the database (Listing A.6).

```
new_user = {
    "email": email,
    "password": generate_password_hash(
        password1,
        method='sha256',
    ),
    "sources": copy.deepcopy(SOURCES_TEMPLATE),
    "sources_metadata":       copy.deepcopy(SOURCES_METADATA_TEMPLATE),
    "keywords": "",
}

with engine.connect() as conn:

    conn.execute(
        Users.insert(),
         new_user,
    )
```

Listing A.6: Sign-Up POST Request Database Changes

Login

For a login API call, the client would make a POST request to the '/auth/login' URI. This would then return if the login was successful or not back to the client.

Sending Request

Listing A.7 is an example of what is sent in the body of a POST request.

```
{
    "email": "chris@hotmail.com",
    "password": "password"
}
```

Listing A.7: Login POST Request Body Example

Return of Request

If the username and password combination are correct, the response in Listing A.8 will be received.

```
Response(
    response=json.dumps({
        "reason": "success"
    }),
    status=200,
    mimetype="application/json",
)
```

Listing A.8: Login POST Request Success Response

Otherwise, the response in Listing A.9 will be received.

```
Response(
    response=json.dumps({
        "reason": "Incorrect Email & Password Combination"
    }),
    status=401,
    mimetype="application/json",
)
```

Listing A.9: Login POST Request Failure Response

Database Changes

There are no database changes for a login request except for adding the user session to the redis database. This will be discussed in section A.2.2.

Session

When a user first logs in, assuming the email and password are valid, a session is created for that user. The session information is stored in a redis database. Redis is an open source, in-memory data store that allows for fast lookups for a given user session [26]. There is a unique GET request under the '/auth/@me' URI which checks if the current user has a session in the redis database and returns this information to the client. The client frequently makes this API call to ensure the user still has access. Otherwise, the user will have to login again and start another session.

Sending Request

No payload body is sent for this GET request.

Return of Request

If there is a session for the user in the redis database, then the response in Listing A.10 will be returned.

```
Response(
    response=json.dumps({
        "id": user["id"],
        "email": user["email"],
    }),
    status=200,
    mimetype="application/json",
)
```

Listing A.10: Session GET Request Success Response

Otherwise an error will be returned (Listing A.11).

```
Response(
    response=json.dumps({
        "reason": "Unauthorized"
    }),
    status=401,
    mimetype="application/json",
)
```

Listing A.11: Session GET Request Failure Response

Logout
When the user clicks the logout button, a POST request to the '/auth/logout' URI is made which pops the user's session from the redis database and redirects them back to the login page.

Sending Request

No payload body is sent for this POST request.

Return of Request

The response in Listing A.12 will be returned by a logout POST request.

```
Response(
    response=json.dumps({
        "reason": "success"
    }),
    status=200,
    mimetype="application/json",
)
```

Listing A.12: Logout POST Request Response

Database Changes

There are no database changes made.

API Resource

There are a total of ten data traversal URIs, they include:

- /api/recommendations_initial
- /api/recommendations_pagination
- /api/process_labels

- /api/papers_initial
- /api/papers_pagination
- /api/papers_label
- /api/invalid_papers_save
- /api/papers_submit
- /api/user
- /api/keywords

Recommendations Initial & Pagination

The '/api/recommendations_initial' GET request receives from the server the first X amount of posts from each enabled source. Currently, X was chosen to be 5 so as to not overload the client with rendering all posts at once. The client keeps track of what page the given source is on. If the user scrolls all the way to the last post, then the client increments the page and makes a GET request to '/api/recommendation_pagination' which will send back the next page of data for that given source.

Sending Request

Listing A.13 is an example of what is sent as part of the payload for the GET request for both the '/api/recommendations_initial' and '/api/recommendations_pagination' URI.

```
{
    'date_before': '2022-07-30T04:00:00.000Z'
    'date_after': '2022-07-05T04:00:00.000Z'
    'keywords': 'machine learning,neural networks'
    'isFilterKeywords': true
    'page': 0
}
```

Listing A.13: Recommendations GET Request Payload Example

Return of Request

Assuming a user session is present in the redis database, then the response in Listing A.14 will be returned for the '/api/recommendations_initial' URI, where posts is a hashmap of sources with their respective html. Listing A.15 is the return request for the '/api/recommendations_pagination' URI.

```
{
    'resultStatus': 'SUCCESS',
    'message': posts
}
```

Listing A.14: Recommendations GET Request Response

```
{
    'resultStatus': 'SUCCESS',
    'message': posts,
    'moreData': false
}
```

Listing A.15: Recommendations Pagination GET Request Response

Database Changes

There are no database changes made with this request.

Process Labels

The '/api/process_labels' URI PUT request is sent once a user labels a given post from a source (Figure A.1). It is updated with either a 0, thumbs-down, or 1, thumbs-up. The user can also reverse this label in which case the same request will be sent but the value will be set as null this time (Figure A.2).

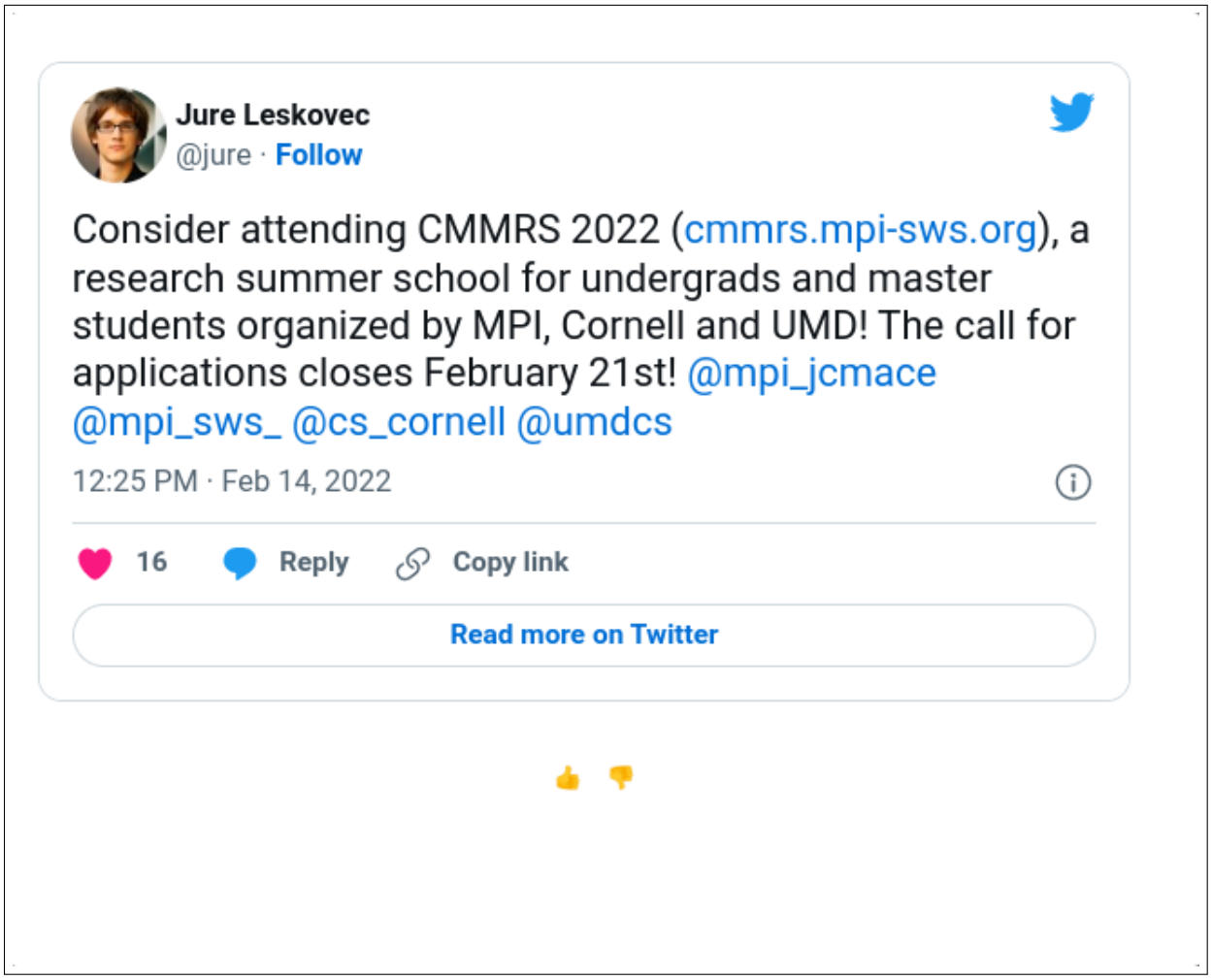


Figure A.1: Unlabelled Post

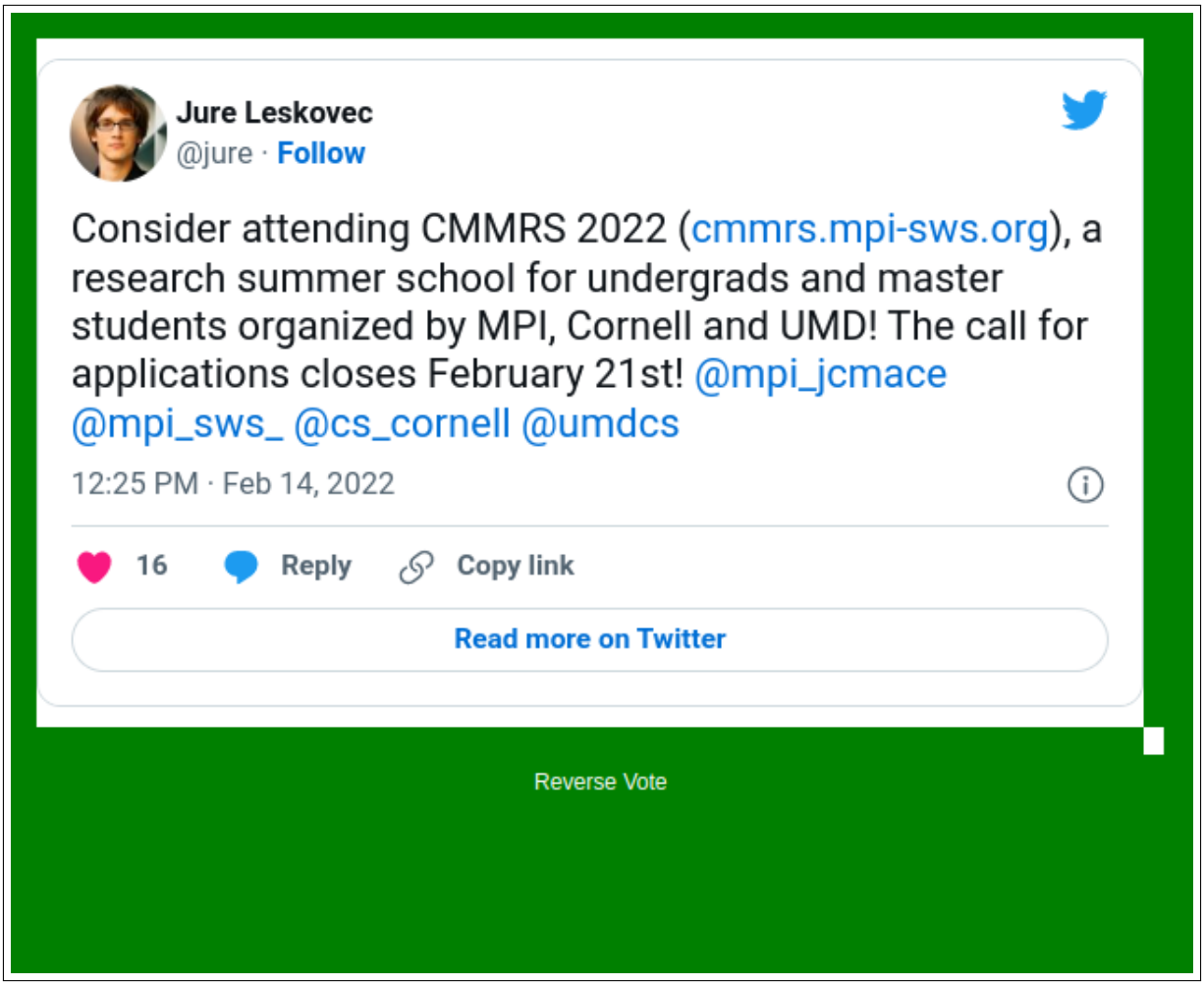


Figure A.2: Labelled Post

Sending Request

Listing A.16 is an example of what will be sent via payload for a PUT request.

```
{
    post2label: {
        id: 5819,
        label: 1,
    }
```

```
}
```

Listing A.16: Label PUT Request Payload Example

Return of Request

Assuming there is a user session present in the redis database, the reponse in Listing A.17 will be sent back.

```
{
    "status": "success"
}
```

Listing A.17: Label PUT Request Response

Database Changes

Listing A.18 details the changes are made to the database with this request.

```
id_ = (
     int(post2label["id"])
     if post2label["id"] is not None
     else post2label["id"]
)

with engine.connect() as conn:

      conn.execute(
          User_Data_Relation.update()
          .values({
              User_Data_Relation.c.label: label
          })
          .where(
              and_(
                  User_Data_Relation.c.user_id == user_id,
                  User_Data_Relation.c.id == id_,
              )
          )
      )
```

Listing A.18: Label PUT Request Database Changes

User

The '/api/user' URI has both a GET and POST request. The GET request simply sends the current user settings to the client and displays which posts are enabled, what the current twitter screen name is and what categories of Arxiv to search for. The POST request sends the same user settings to be updated on the database, using the backend, once the user alters them.

The GET request does not have anything to send in the payload, however an example of what would be sent in the POST request payload can be found in Listing A.19.

Sending Request

Listing A.19 is an example of what is in the body of the payload when sending a POST request.

```
{
    arxiv: {enabled: true, ...}
    medium: {enabled: true}
    pmlr: {enabled: true}
    reddit: {enabled: true}
    twitter: {enabled: true, username: "chiDeltaMu"}
    youtube: {enabled: true}
}
```

Listing A.19: User POST Request Payload Example

Return of Request

GET

Assuming there is a user session in the redis database, the response in Listing A.20 will be returned.

```
{
   'resultStatus': 'SUCCESS',
   'userInfo': data
}
```

Listing A.20: User GET Request Response

Where ‘data’ is a hashmap of the user’s settings for each source.

POST

Assuming there is a user session in the redis database and the provided twitter screen name is valid, then the response in Listing A.21 will be sent.

```
{
    'status': "Success"
}
```

Listing A.21: User POST Request Success Response

If the screen name is not valid, then the response in Listing A.22 will be sent back.

```
{
    'status': "Warning, incorrect twitter screen name!"
}
```

Listing A.22: User POST Requeset Failure Response

Database Changes

Listing A.23 outlines the changes that are made to the database with the POST request.

```
with engine.connect() as conn:

   conn.execute(
       Users.update()
       .values({
           Users.c.sources: data
       })
       .where(
           Users.c.id == user_id
       )
   )
```

Listing A.23: User POST Request Database Changes

Keywords

The ‘/api/keywords‘ URI has both a GET and POST request. The GET request sends the current keywords to the client, which then displays them (Figure 1.18), while the POST

request will send the updated keywords to the database if the user has altered them. The GET request does not have anything to send in the payload, however an example of what would be sent in the POST payload can be found below.

Sending Request

Listing A.24 is an example of the body for the payload of a POST request.

```
{

    keywords: "machine learning\nneural networks"

}
```

Listing A.24: Keywords POST Request Payload Example

Return of Request

GET

Assuming there is a user session in the redis database, then the response in Listing A.25 will be returned.

```
{
   "resultsStatus": "SUCCESS",
   "message": keywords,
}
```

Listing A.25: Keywords GET Request Response

Where 'keywords' is a newline delimited string of the user's keywords.

POST

Assuming there is a user session present in the redis database, then the response in Listing A.26 will be returned:

```
{
    "status": "Success"
}
```

Listing A.26: Keywords POST Request Response

Database Changes

The following changes will be made to the database with the POST request:

```
with engine.connect() as conn:

    conn.execute(
        Users.update()
        .values({
            Users.c.keywords: keywords
        })
        .where(
            Users.c.id == user_id
        )
    )
```

Papers Submit

The '/api/papers_submit' URI has a POST request. The POST request accepts a newline delimited string of arxiv.org urls; afterward, these papers are added to the database and saved as positively labeled data. The area to submit these urls is shown in Figure 1.11.

Sending Request

Listing A.27 is an example of the body for the payload of a POST request.

```
{
    "urls": "arxiv.org/abs/1234.12345\narxiv.org/abs/3241.56435",
}
```

Listing A.27: Papers Submit POST Request Payload Example

Return of Request

POST

Assuming there is a user session present in the redis database, then the response in Listing A.28 will be returned. The 'invalid_papers' variable is a newline delimited string of urls the backend was unable to process.

```
{
    'resultStatus': 'SUCCESS',
    'papers': papers,
    'invalid_papers': invalid_papers,
}
```

Listing A.28: Papers Submit POST Request Response

Database Changes

Listing A.29 details the changes that will be made to the database with the POST request.

```
with engine.begin() as conn:

    for x in data:

        out = conn.execute(
            data_table.insert()
            .values(
                x,
            )
        )

        conn.execute(
            papers_table.insert()
            .values(
                {
                    "data_id": out.lastrowid,
                    "user_id": user_id,
                    "label": 1,
                }
            )
        )

    for id_ in primaryKeysPresent:

        conn.execute(
            papers_table.insert()
            .values(
                {
                    "data_id": id_,
                    "user_id": user_id,
                    "label": 1,
```

```
                    }
                )
            )
```

Listing A.29: Papers Submit POST Request Database Changes

The variable ‘data‘ represents all of the new papers that were not present in the ‘data‘ table, while the ‘primaryKeysPresent‘ variable represents a list of the papers that were already present in the ‘data‘ table.

Papers Invalid URLs

The ‘/api/invalid_papers_save‘ URI has a PUT request. The PUT request accepts a newline delimited string of the saved invalid urls.

Sending Request

An example of the body for the PUT request payload is shown in Listing A.30.

```
{
    "invalid_urls": "arxi.com\narxiv.org/abs/12.4356"
}
```

Listing A.30: Papers Invalid PUT Request Payload Example

Return of Request

PUT

Assuming there is a user session in the redis database, then the response in Listing A.31 will be returned.

```
{
    "status": "success"
}
```

Listing A.31: Papers Invalid PUT Request Response

Database Changes

Listing A.32 details the changes will be made to the database with the PUT request.

```
query = sqlalchemy.text("""
        UPDATE users
        SET invalid_papers = :invalid_urls
        WHERE
            id = :user_id
    """).bindparams(
        invalid_urls=invalid_urls,
        user_id=user_id,
    )

with engine.connect() as conn:

    conn.execute(query)
```

Listing A.32: Papers Invalid PUT Request Database Changes

Papers Label

The '/api/papers_label' URI has a PUT request. The PUT request accepts a hash-map of the paper id and the label. Figure A.3 shows where a user would re-label a given paper.

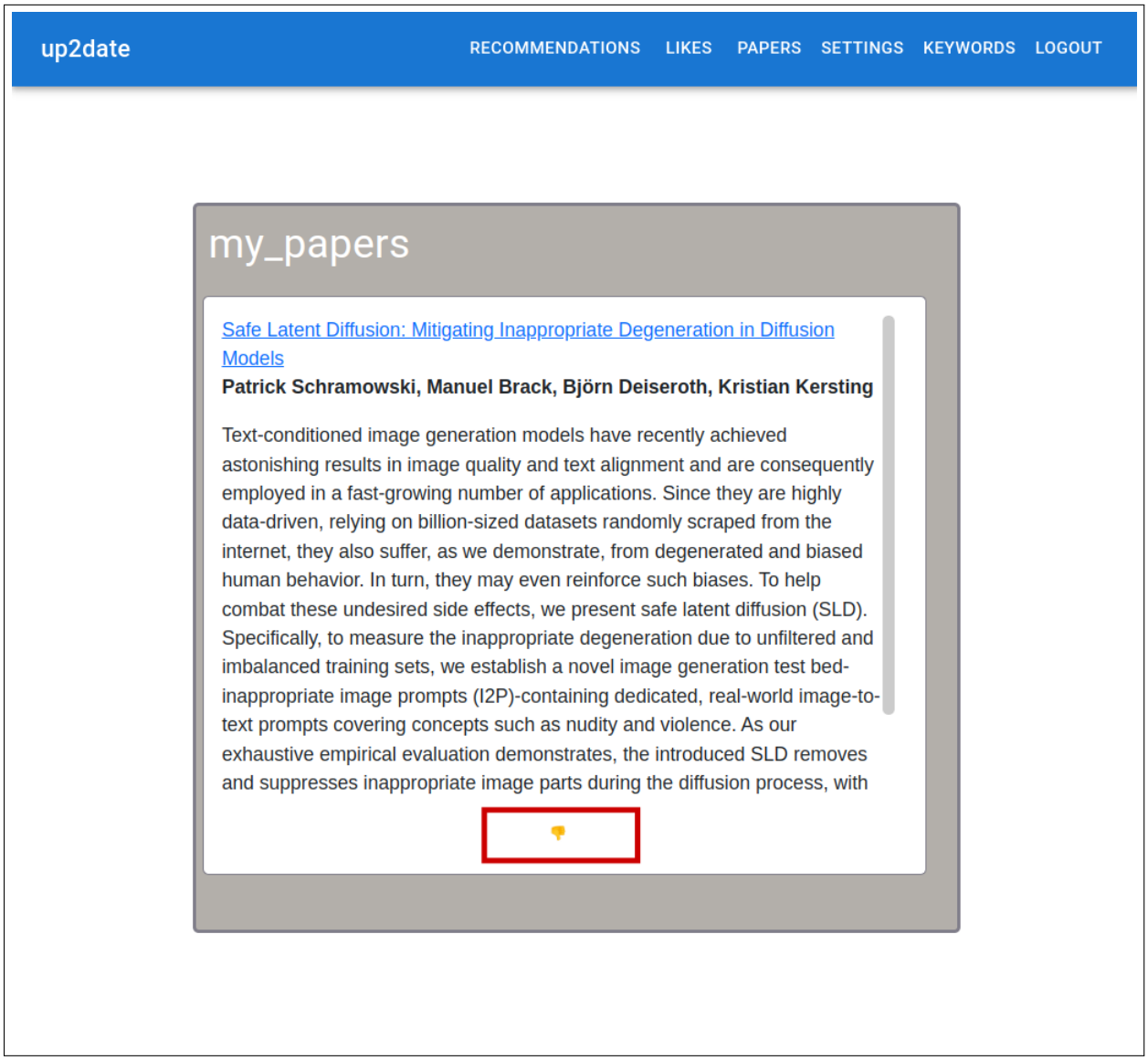


Figure A.3: Papers Page - Labeling a Paper

Sending Request

Listing A.33 is an example of the body for the PUT request payload.

```
{
    id: 1453,
    label: 0,
}
```

Listing A.33: Papers Label PUT Request Payload Example

Return of Request

PUT

Assuming there is a user session in the redis database, then the request in Listing A.34 will be returned.

```
{
    'resultStatus': 'SUCCESS',
}
```

Listing A.34: Papers Label PUT Request Response

Database Changes

Listing A.35 details the changes that will be made to the database with the PUT request.

```
query = sqlalchemy.text("""
        UPDATE papers
        SET label = :label
        WHERE
            user_id = :user_id
            AND
            id = :id
        ;
    """).bindparams(
        label=label,
        user_id=user_id,
        id=id,
)

with engine.connect() as conn:

    conn.execute(
```

```
        query
    )
```

Listing A.35: Papers Label PUT Request Database Changes

Papers Pagination

The '/api/papers_pagination' URI has a GET request. The GET request returns the next set of papers to display after a user has scrolled to the end of the list.

Sending Request

Listing A.36 is an example of the body for the GET request payload.

```
{
    'page': 0,
    'n_labelled': 0,
}
```

Listing A.36: Papers Pagination GET Request Payload Example

Return of Request

GET

Assuming there is a user session in the redis database, then the response in Listing A.37 will be returned.

```
{
    'resultStatus': 'SUCCESS',
    'message': papers,
    'moreData': moreData,
}
```

Listing A.37: Papers Pagination GET Request Response

The variables 'papers' and 'moreData' are the next set of papers to display and a boolean variable indicating if there is more data to display or not, respectively.

Database Changes

There are no database changes made with this resource call.

A.2.2 Sessions

To facilitate a time persistent access to the application, a user session was implemented. First a flask application is instantiated (Listing A.38).

```
from flask import Flask

app = Flask(
       __name__,
       static_url_path='',
       static_folder='client/build',
)
```

Listing A.38: Instantiate Flask Application

The application configuration must be written to include the use of sessions (Listing A.39).

```
class ApplicationConfig(object):

   SESSION_TYPE = "redis"
   SESSION_PERMANENT = False
   SESSION_USE_SIGNER = True
   SESSION_REDIS = redis.from_url(
       os.environ.get('REDIS_URL')
    )

app.config.from_object(ApplicationConfig)
```

Listing A.39: Flask Configuration

Afterwards the Session function is applied to the app (Listing A.40).

```
from flask_session import Session

Session(app)
```

Listing A.40: Applying Session to Application

To add a user session in the 'auth/login' POST method, the server code is executed as in Listing A.41.

```
from flask import session

data = request.get_json(silent=True)

email = data.get("email")
password = data.get("password")

user = table_search_where(
    Users,
    Users.c.email,
    email,
    isFetchOne=True,
)

if user is not None:

    if check_password_hash(user.password, password):

        session["user_id"] = user["id"]
```

Listing A.41: Adding a User Session

After this, the user's id is now in the redis database and currently has a session. When the user logs out of the application, the server simply pops the user id from the redis database (Listing A.42).

```
from flask import session

session.pop("user_id")
```

Listing A.42: Removing User Session from Redis Database

A.3 Database

The data that was planned to be stored includes information about users (email, hashed password, sources, sources_metadata), posts (text, html, source) from a certain source and user labels on said posts. This type of data is structured; thus, a MySQL server was chosen to store this data. A total of seven tables were used in the database. They are as follows:

- users
- data

- user_data_relation
- negative_data
- logs
- master_metadata
- papers

A.3.1 Tables

This section will outline the seven tables that were used in this database.

Users Table

The users table keeps a record of the user's email, hashed password, sources and sources_metadata. The sources column is a JSON which tracks what sources a user would like to display in the application and have data collected for. The sources_metadata is a JSON which tracks when a source was last updated with new posts for the given user.

Sample Row

An example of a row from the users table is shown in Listing A.43 (note the hashed password is not included for security reasons).

```
{
    "id": 1,
    "email": "chris@hotmail.com",
    "sources": {
        "pmlr": {
            "enabled": true
        },
        "arxiv": {
            "enabled": true,
            "categories": {
                "cs.AI": true,
                ...
                "cond-mat.stat-mech": true
            },
```

```
            "category_map": {
                "cs.AI": "Artificial Intelligence",
                ...
                "cond-mat.stat-mech": "Statistical Mechanics"
            }
        },
        "medium": {
            "enabled": true
        },
        "reddit": {
            "enabled": true
        },
        "twitter": {
            "enabled": true,
            "username": "chiDeltaMu"
        },
        "youtube": {
            "enabled": true
        }
    },
    "sources_metadata": {
        "pmlr": {
            "last_primary_id": 76932
        },
        "arxiv": {
            "neural networks": "Fri Jun 03 02:34:52 2022",
            "machine learning": "Fri Jun 03 02:34:52 2022"
        },
        "medium": {
            "last_id": 79690
        },
        "reddit": {
            "neural+networks": 1654561276.4528103,
            "machine+learning": 1654561273.6200736
        },
        "twitter": {
            "jure": {
                "since_id_str": "1529313966370869249"
            },
            ...
            "DeepMind": {
                "since_id_str": "1532006262728278018"
            }

        },
```

```
        "youtube": {
            "neural networks": "Tue Jun 07 00:21:58 2022",
            "machine learning": "Tue Jun 07 00:21:56 2022"
        }
    }
}
```

Listing A.43: Users Table Sample Row

For example, the keyword of "neural networks" for the source "youtube" was last downloaded on Tuesday June 7th 00:21:58 2022.

Data Table

The data table keeps a record of the source, text, html, timestamp, papers with code link and data id. The papers with code link, if available, provides the address to paperswithcode.com for the applicable paper. The 'data_id' is the id of the post as given by its own source. For example, twitter provides a 64-bit unsigned integer as its id.

Sample Row

```
    {
    "id": 12980,
    "data_id": "1493275276200984576",
    "source": "twitter",
    "html": "<blockquote class=\"twitter-tweet\"><p lang=\"en\" dir=\"ltr
    \">Consider attending CMMRS 2022 ...",
    "text": "Consider attending CMMRS 2022 (https://t.co/8v2ed82BiE), a
    research summer school for undergrads ...",
    "created_at": "2022-02-14 17:25:44",
    "pwc_link": null
}
```

Listing A.44: Data Table Sample Row

User Data Relation Table

The user_data_relation table keeps a record of the user id, data id, label, and prediction. The user id refers to the user for which this record references (which is the primary key from the users table). The data id refers to the primary key for the row in the data table

for which this record references. The label and prediction is assigned by the user and trained machine learning model respectively. If this table was not created, then the labels, predictions, and user id would have had to be tacked on to each row in the ‘data‘ table and would likely result in redundant information. For example, if two users had a reference to the same tweet, then there would be two rows in the ‘data‘ table referencing this tweet. The only difference between these two rows would be the prediction, label, and user id columns. The relationship between the ‘data‘ table and the ‘user_data_relation‘ table is thus one-to-many.

Sample Row

```
{
    "id": 1,
    "data_id": 35,
    "user_id": 1,
    "label": null,
    "prediction": null,
}
```

Listing A.45: User Data Relation Table Sample Row

Papers Table

The papers table is similar to the user_data_relation table in that it has a data id, user id, and label which refer to the primary key in the data table, the primary key for the given user, and the label for the paper itself respectively. Normally when a user adds a paper via the application, a label of one is applied; however, it is possible the user will later dislike the paper and a label of zero must be applied.

Sample Row

```
{
    'id': 143,
    'data_id': 94033,
    'user_id': 3,
    'label': False
}
```

Listing A.46: Papers Table Sample Row

Negative Data Table

The negative_data table keeps a record of the data id, source, text, and timestamp. The data id is given by the source. The data in this table is deemed to be a negative sample for any given user and a specific source. For example, in this table are random tweets which can be used as negative training points for twitter (and any given user).

Sample Row


```
{
    "id": 621,
    "data_id": "1525982866865369101",
    "source": "twitter",
    "text": "went to Florida for the weekend \ud83d\ude0a https://t.co/1
    UhBMdReYX",
    "created_at": "2022-05-15 23:34:01"
}
```


Listing A.47: Negative Data Table Sample Row

Logs Table

The logs table keeps a record of the user id, source, message, timestamp and arguments. The message is the error received when trying to get new data for a given user and source and the arguments are the function arguments that were passed into the data collection function. The logs table can be leveraged to assess what errors may be present in the data collection process and perform root cause analysis to fix it.

Sample Row


```
{
    "id": 7,
    "user_id": 2,
    "source": "youtube",
    "message": "cannot unpack non-iterable NoneType object\nTraceback (
    most recent call last):...",
    "created_at": "2022-05-31 23:15:16",
    "arguments": "{'args': [{'random walks': 'Tue May 31 23:15:16 2022', '
    graph diffusion': 'Tue May 31 23:15:15 2022', ..."
```

```
}
```

Listing A.48: Logs Table Sample Row

Master Metadata Table

The master_metadata table has only one row and one column which stores a JSON that keeps a record of when certain sources were last collected. Sources such as towards-data-science are updated using this method as it is more efficient to collect all new data from the source, add it into our database and then search the database based on keywords for each user.

Sample Row

```
{
    "id": 1,
    "sources_metadata": {
          'core': {'previous_idx': 3600},
          'pmlr': {'last_volume_scraped': 'v190'},
          'medium': {'last_day_scraped': '2022-11-14'},
          'medium_negative': {'last_day': '2016-01-30'},
     }
}
```

Listing A.49: Master Metadata Sample Row

A.3.2 Foreign Keys

There are a total of five foreign keys that are used across all tables to ensure data integrity. Two of these are from the user_data_relation table to reference the primary id from the data and users tables. Another two are from the papers table to reference the primary id from the data and users table. The last one is present in the logs table and it references the users table primary id.

As well, with these foreign keys is a requirement to cascade on delete. This means that when a row from a table (which is referenced by another table via foreign key) is deleted, all other rows with said foreign key are also deleted. A visualization of all tables in the database can be found in Figure A.4. Arrows indicate the ForeignKey Relationships.



Figure A.4: Database Schema

A.3.3 Accessing the Database

In order to access the database for SQL queries, SQLAlchemy was used [27]. SQLAlchemy was chosen as both the backend of the application and modeling was written using Python. SQLAlchemy enabled streamlined access to the database. Setting up a database engine can be done using only a few lines of code. For example, if a user does not require SSL to login, a connection can be made using the code in Listing A.50.

```
engine = create_engine(
    database_url,
)
```

Listing A.50: Creating Engine wihtout SSL

As well, if a user is required to use SSL to login, then it can be done using the code in Listing A.51.

```
engine = create_engine(
    database_url,
    connect_args={
        'ssl': {
            'ca': os.path.join(
                os.getcwd(),
                'amazon-rds-ca-cert.pem',
            )
        }
    },
)
```

Listing A.51: Creating Engine with SSL

Once the engine has been initialized, it is trivial to write queries and return the associated data from the database (Listing A.52).

```
with engine.connect() as conn:

    x = conn.execute(
        Data.select(),
    ).fetchall()
```

Listing A.52: Sample SQLAlchemy Query

A.3.4 Database Users

There are a total 3 database users that are involved in accessing the database for this application.

- admin
- datafetch_user
- app_user

Admin

Admin credentials are only stored on my local machine and used from my local machine. This ensures that the EC2 instance and web application do not have administrative privileges for the database.

Datafetch User

The datafetch_user was created to allow for the remote server to access the database to add new data to the database and update predictions. This user was given the following privileges shown in Listing A.53.

```
'GRANT SELECT, INSERT, UPDATE ON `user_data_db`.* TO `datafetch_user`@`<
    hostname>`'
```

Listing A.53: DataFetch User Database Grants

The user has nearly all privileges except for deleting data from the database. This is because no tasks on the remote server require this privilege.

App User

The app_user is used for the backend of the application to make calls to the database. The following privileges were given to the user as shown in Listing A.54.

```
GRANT SELECT ON `user_data_db`.* TO `app_user`@`%`

GRANT UPDATE (`label`) ON `user_data_db`.`user_data_relation` TO `
    app_user`@`%`
```

Listing A.54: App User Database Grants

The user was given the permission to read all data via the first statement above, while the second statement gives them the permission to update the label of a post when a user does so using the client. The backend has no requirements to delete or insert data which

is why there is no permission given to them. As the application user is not on the same virtual network as the database nor does it have a constant IP address (as is for the admin), the user is required to connect via SSL. This was done when the user was first created as shown in Listing A.55.

```
CREATE USER app_user REQUIRE SSL;
```

Listing A.55: Creating the App User

A.3.5 Normalized vs. Denormalized Tables

As mentioned previously (Section A.3.1), the post data and user labels (and predictions) were normalized to ensure there would not be redundant storage of a post's text, html, papers with code link, created at date, and data id. This normalization resulted in creating the 'data' and 'user_data_relation' tables. When sending data to the client for posts to recommend, the backend must make the following query to the database for the user's recommendations:

```
SELECT * FROM user_data_relation
LEFT JOIN data
ON user_data_relation.data_id = data.id
WHERE
  user_data_relation.user_id = <user_id>
  AND
  user_data_relation.prediction > 0.5
LIMIT 5
OFFSET <page>
;
```

Listing A.56: Query to Retrieve Data

This requires a left join of the 'data' table onto the 'user_data_relation' table which can be computationally expensive. Thus, in many cases, it is common to de-normalize these two tables into one. This reduces the computational cost of joins while increasing the storage cost of redundant data.

A.4 Remote Server

A remote server in the form of an AWS (Amazon Web Services) EC2 (Elastic Cloud Compute) server was used to fetch user data, train models, apply inference on fetched user data and fetch negative data.

A.4.1 Fetching User Data

Fetching user data uses a nested loop in which the outer loop iterates over all users while the inner loop iterates over all sources for the given user. If the source is enabled for the given user, then the data collection function for that source is passed the user's sources_metadata and returns any new posts that match the search criteria. The search criteria usually contains a hashmap of keywords and the last time they were searched for. In cases like twitter, it contains the since_id of the last tweet for each followee of the given user. In both of these cases, keeping track of the last time a search was made or the last id reduces both the time spent searching for relevant posts and the amount of redundant data that could be added to the database. A complete flowchart of the steps taken to fetch user data can be found in Figure C.1.1.

For certain sources like towards-data-science (Figure C.1.2) and PMLR (Figure C.1.3), it is not efficient to scrape on a per user basis, as there is no utilization of search queries using keywords. Thus, all new posts from these sources are added to the database. When a given user-source search function is run it will simply search the database for all source posts greater than the last recorded id, which is stored in the user's sources_metadata, that also match the user's keywords.

Model Training

For training models, a loop is used to train one model per user. There are two possible scenarios that are used to provide recommendations to the user and they depend upon the amount of positive user-labeled data that is present in the database. If the number of positively labeled data points is less than 100, then no model inference is applied to the data and keyword matching is only used. In this case, all fetched data is shown to the user. If the number of positively labeled data points is greater than 100, then a machine learning model is trained. In this case, only positive predictions by model inference are shown to the user.

Once it is decided if a machine learning model will be trained, then the data is loaded, tokenized, and passed into the model for training. Once this is complete, both the trained tokenizer and model are saved to storage for the given user.

Model Inference

Applying inference on fetched data also requires a loop for the users. For a given user, if there is a saved model then inference is applied. First the data to be inferred is tokenized

using the trained tokenizer and then this data is passed into the machine learning model. The predictions of this model are used to update the "prediction" column in the database.

Fetching Negative Data

As we are usually limited by the number of positively labeled samples when training a model, a method to ensure there is always enough negatively labeled samples for each source was devised. A nested loop of sources and users, with the outer and inner loops comprising the sources and users respectively, calculates the difference between the number of positive and negative samples for a given user and source pair. A max difference is recorded for each source. If this difference is greater than zero, then a function is used to collect the number of data points equal to that max difference. This ensures that each model will never be limited by the number of negative samples as these are simple enough to obtain.

Dockerization

In order to facilitate a seamless transition from running these functions locally to the cloud, Docker was used. Docker is a software package that uses operating system level virtualization to deliver software in packages called containers [28]. A Dockerfile was created that would store all necessary functionality including a virtual environment, webdriver for use with selenium, and repository code.

To create containers to run the images created from the Dockerfile, docker-compose.yml files were written. These specify all necessary resources for the container such as the docker image, ports mappings, volume mappings and more. The docker-compose.yml used for these containers can be found in Listing A.57.

```
version: "3"
services:
  fetch_user_data:
    container_name: fetch_user_data
    image: aws_tasks
    volumes:
    - ./logs/:/datafetch/logs/
    - ./data/:/datafetch/data/
    network_mode: host
    command: ["python", "aws_scripts/aws_fetch_data.py"]
  fetch_negative_data:
    container_name: fetch_negative_data
```

```
    image: aws_tasks
    volumes:
    - ./logs/:/datafetch/logs/
    - ./data/:/datafetch/data/
    network_mode: host
    command: ["python", "aws_scripts/aws_fetch_negative_data.py"]
  train_ml_models:
    container_name: train_ml_models
    image: aws_tasks
    volumes:
    - ./logs/:/datafetch/logs/
    - ./data/:/datafetch/data/
    network_mode: host
    command: ["python", "aws_scripts/train_ml_models.py"]
  apply_ml_inference:
    container_name: apply_ml_inference
    image: aws_tasks
    volumes:
    - ./logs/:/datafetch/logs/
    - ./data/:/datafetch/data/
    network_mode: host
    command: ["python", "aws_scripts/apply_ml_inference.py"]
```

Listing A.57: Docker Compose YAML

To create these containers, the following command in Listing A.58 must be run on the command line.

```
sudo docker compose up --no-start
```

Listing A.58: Creating Containers

This will create the four containers in the YAML file. Afterward, to re-run any of these jobs, the container can simply be restarted as in Listing A.59.

```
sudo docker restart <container_name>
```

Listing A.59: Restarting Docker Container

Cronjobs

Cronjobs were utilized in order to download user data and train user models. Table A.1 details the jobs, their description and the frequency at which they are performed.

Job	Description	Frequency
Fetch User Data	Loop over each user and each source to download new data	Once per day
Train ML Models	Loop over each user and train a new model	Every hour
Apply ML Inference	Loop over each user and apply inference to unlabeled data points	Every hour (after model training)
Fetch Negative Data	Loop over each source and determine the maximum difference between the number of positive and negative data points with respect to all users	Every Week

Table A.1: Cronjobs for Data Collection and Model Training

In order to execute each of these jobs, the code in Listing A.60 was added to the remote server's crontab file.

```
# m h  dom mon dow   command

0 * * * * sudo docker restart train_ml_models && sudo docker restart
   apply_ml_inference

0 5 * * * sudo docker restart fetch_user_data

0 9 * * 6 sudo docker restart fetch_negative_data
```

Listing A.60: Additions to crontab file

Appendix B

Data

The data that is used for modeling in this application is textual. A total of 6 sources have been implemented and include twitter, reddit, pmlr, arxiv, towards-data-science, and youtube. Some of these sources have an API (Application Programming Interface) while others do not; Table B.1 details this.

In the cases where an API is available, the process to collect data is fairly simple. The only real necessity is to acquire an API token (in the cases of twitter, reddit, and youtube). Arxiv does not require an API token to utilize their API. For PMLR (Proceedings of Machine Learning) and towards-data-science the requests library was used to collect data. Table B.1 also details which sources are searched on a "per user" basis. This means that whenever data is collected from this source, it is done specifically with the intent to collect data for one user. Otherwise, all new data is collected, added to the database and then each user's keywords can be matched to the data in the database for recommendations.

Source	**API?**	**Per User Search?**
twitter	Yes	Yes
reddit	Yes	Yes
arxiv	Yes	Yes
pmlr	No	No
towards-data-science	No	No
youtube	Yes	Yes

Table B.1: Source Details

B.1 Data Collection

This section outlines the methods of data collection for each source.

B.1.1 Twitter

Method of Collection

In order to leverage twitter's API, the python library tweepy was used [29]. To begin with, a tweepy api object must be created using the api tokens which include the consumer key, consumer secret, access token and access token secret (Listing B.1).

```
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)

api = tweepy.API(
   auth,
   wait_on_rate_limit=True,
)
```

Listing B.1: Tweepy API

With this api object, one can search for all tweets made by a specified screen name after a given tweet id (Listing B.2).

```
user_statuses = tweepy.Cursor(
       api.user_timeline,
       screen_name=screen_name,
       since_id=since_id,
       tweet_mode="extended",
).items()
```

Listing B.2: Retrieving User Statuses using Tweepy

Each status text is iterated upon to determine if it matches the user's specified keywords or not. If not, the tweet is discarded. The general approach taken is to collect tweets, that match the given keywords, from a user's followees on twitter.

Data Format

Tweepy returns their own version of a tweet object from the api query. Embedded within this object is a JSON object. This JSON object contains information such as the

time of creation, id, and text. The html of the embedded tweet can also be obtained using another call to the api (Listing B.3).

```
html = api.get_oembed(
    "https://twitter.com/" +
    f"{screen_Name}" +
    f"/status/{id}"
)["html"]
```

Listing B.3: Retrieving Embeded Tweets

One problem with this is that the embedding of the tweet shows replies, but the text extracted from the tweet does not. For example, let's consider the tweet shown in Figure B.1.



Figure B.1: Sample Tweet

When embedded, it actually shows the following in Figure B.2.

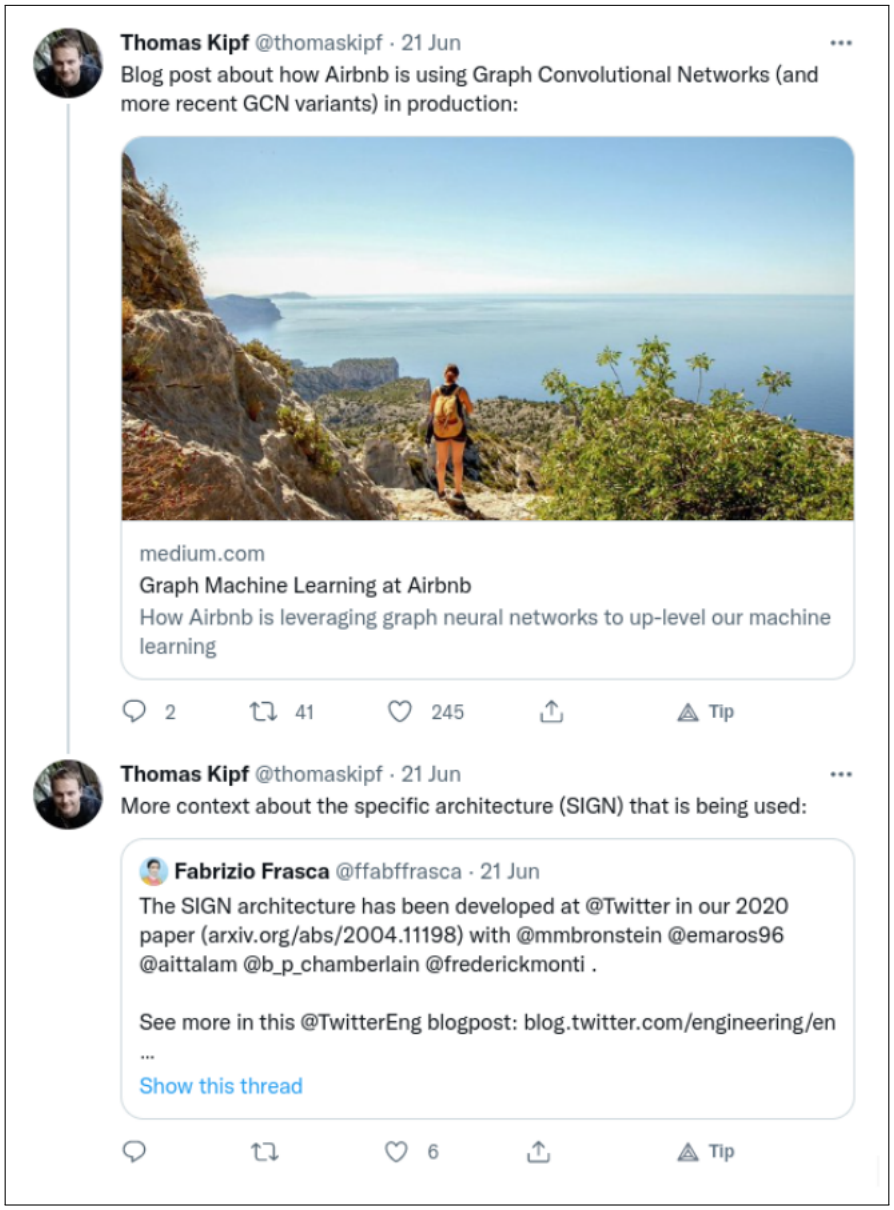


Figure B.2: Sample Tweet with Reply

When extracting the "text" from this tweet JSON, we are only provided with the following text:

"More context about the specific architecture (SIGN) that is being used: https://t.co/rZaXlHzShC"

This is missing the quoted text ("The SIGN architecture ...") and the replied text ("Blog post about ..."). In order to alleviate this, the replies to this tweet are stitched together to the original text of the tweet which is then returned as the complete text. The stitched text of the tweet is as follows:

"Blog post about how Airbnb is using Graph Convolutional Networks (and more recent GCN variants) in production: https://t.co/rAdyCAs8g7 More context about the specific architecture (SIGN) that is being used: https://t.co/rZaXlHzShC The SIGN architecture has been developed at @Twitter in our 2020 paper (https://t.co/xkLNLZ6zmm) with @mmbronstein @emaros96 @aittalam @b_p_chamberlain @frederickmonti . See more in this @TwitterEng blogpost: https://t.co/SylhKRcQtO"

A sample of a twitter JSON post can be found in Appendix C.2.1.

Collection Process

Twitter data is collected on a per user basis as each user has their own followees and keywords which they would like to search for. When starting this collection process for a given user, the user's twitter metadata and keywords are required. The twitter metadata is a hashmap containing each of the user's current followees as keys and the last tweet id that was collected for the followee as a value. A loop is created for each followee and the tweepy API is used to search for all tweets since the given last tweet id. The appropriate information is extracted from each tweet and added to a list. The metadata is updated with the newest tweet id for each followee. The tweets are then filtered via user keyword matches . Finally, the collected data and metadata are added and updated on the database respectively. A flowchart of this process can be found in Appendix C.1.4.

B.1.2 Reddit

Method of Collection

In order to collect reddit data, the python requests library is used in conjunction with reddit API tokens. To leverage the API, you must get the appropriate header to pass when

making a call to the api. This header includes the app name and the authorization token (Listing B.4).

```
auth = requests.auth.HTTPBasicAuth(
   REDDIT_CLIENT_ID,
   REDDIT_SECRET_TOKEN,
)

# here we pass our login method (password), username,
# and password
data = {'grant_type': 'password',
        'username': REDDIT_USERNAME,
        'password': REDDIT_PASSWORD}

# setup our header info, which gives reddit a
# brief description of our app
headers = {'User-Agent': f'{REDDIT_APP_TITLE}/0.0.1'}

# send our request for an OAuth token
res = requests.post(
        'https://www.reddit.com/api/v1/access_token',
        auth=auth,
        data=data,
        headers=headers,
)

# convert response to JSON and pull access_token value
TOKEN = res.json()['access_token']

# add authorization to our headers dictionary
headers = {
   **headers,
   **{'Authorization': f"bearer {TOKEN}"}
}
```

Listing B.4: Reddit API

Once the header is created, a call to the api may be made with the requests python library (Listing B.5). The filter can refer to relevance or top posts, while the timeline can be last month, year, all-time, etc. The params term specifies how many posts you would like to limit the request to return .

```
res = requests.get(
    "https://oauth.reddit.com/r/all/search.json?q=" +
    term +
    "&sort=" +
```

```
    filter +
    "$t=" +
    timeline,
    headers=headers,
    params={'limit': limit},
)
```

Listing B.5: Retrieving Reddit Data

Data Format

The object returned from the reddit API is a JSON. From the JSON object, the creation time, text, html and id can be extracted. The text is a concatenation of the title and the body of the post. The html of the embedded post is constructed using the link to the post. A sample of a reddit JSON post can be found in Appendix C.2.2.

Collection Process

Reddit data is collected on a per user basis similarly to twitter as each user has their own reddit metadata and keywords. The reddit metadata contains a hashmap of the keywords as keys and the last time they were searched for, using the reddit API, as values.

A loop over each of these keywords is performed. If the keyword has not been searched before, then a nested loop is traversed over for filters "relevance" and "top". The keyword is then searched for with filters "hot" and "new". The idea here being that keywords you haven't searched for before you want all possible information while for other keywords you just want the latest and greatest. The current time is then updated on the hashmap for each keyword to indicate the last time it was searched for and is updated on the database. A flowchart of this process can be found in Appendix C.1.5.

B.1.3 Arxiv

Method of Collection

Arxiv provides a method to access their data via an API. Their API does not require a token. Data can be received via the url found in Listing B.6.

```
http://export.arxiv.org/api/query?search_query=all:<keyword(s)>&start=0&
    max_results=<max_results>
```

Listing B.6: Arxiv API URL

Once the url has called the API, the data is returned in Atom 1.0 format. It is suggested to decode the data using the urllib library (Listing B.7).

```
url = 'http://export.arxiv.org/api/query?search_query=all:electron&start
    =0&max_results=1'
data = urllib.request.urlopen(url)
decoded_data = data.read().decode('utf-8')
```

Listing B.7: Retrieving Arxiv Data

The decoded data is a hashmap which contains a list of the different papers from the search query that can be iterated over.

Data Format

Each paper is stored as a hashmap. From this hashmap, the title, abstract, authors, publish date, and id can be extracted. The title and abstract are concatenated to create the text. The html of the paper is created with a few basic tags from the text, authors, arxiv.org link, and a papers with code link. A sample of the decoded data hashmap can be found in Appendix C.2.4.

Collection Process

The keywords for the user are iterated over and a search query string is created for each. If the keyword is not in the user's arxiv metadata then papers are searched for using the query and ordered by relevance. A second query ordered by date is searched for irrespective of the keyword being in the metadata or not. The last searched date from the metadata is supplied to this second query if the keyword is in the metadata, otherwise a date of two weeks prior is supplied. After this, a papers with code link, if available, is added to each item. Finally, the metadata for each keyword is updated with the current date. The metadata is then also updated on the database. A flowchart of this process can be found in Appendix C.1.7.

B.1.4 PMLR

Method of Collection

PMLR is available via an RSS feed. This data is processed using the python requests library and any new papers are identified. As PMLR provides no API access, all new PMLR articles are collected and added to the database. After this, each user's keywords are matched with what is in the database.

Data Format

The RSS file is in XML format. This file is easily traversed for each volume, and paper, via the beautiful soup python library. A sample of the PMLR XML feed can be found in Appendix C.2.3.

Collection Process

In order to collect PMLR data, the RSS XML file is first analyzed. Using the PMLR metadata which contains the last volume collected, all new volumes that are present in the RSS feed are determined. Each of these volumes is traversed, processed and the papers are added to the database. As PMLR does not provide a unique id for each paper, an md5 hash of the text is used as the data id. Once this is complete, the PMLR metadata is updated with the latest volume and is updated on the database. A flowchart of this process can be found in Appendix C.1.6.

B.1.5 Towards Data Science

Method of Collection

Towards-data-science articles are accessible only via web portal using their archive at medium.com/towards-data-science/archive. There is no API that can be used to search for blog posts. Thus, the python requests library is used to access these posts.

Data Format

As mentioned, the blog posts are only available via web browser and thus the format is HTML. A sample of a towards-data-science blog post HTML can be found in Appendix C.2.6.

Collection Process

As towards-data-science is unable to be leveraged via API for specific user searches, all blog posts are collected for each day and added to the database. There is no unique id given by towards-data-science for each blog post, thus an md5 hash of the text is used to create a data id for each post. After this, each user's keywords are compared with new posts and matches are recorded. A flowchart of this process can be found in Appendix C.1.2.

B.1.6 Youtube

Method of Collection

Youtube data can be accessed through the use of an API which can be attained via Google Cloud Platform (Listing B.8).

```
from googleapiclient.discovery import build

def youtube_authenticate_api_key(api_service_name, api_version, api_key):
    return build(api_service_name, api_version, developerKey=api_key)


api_key = os.environ.get("YOUTUBE_API_KEY")

youtube = youtube_authenticate_api_key(
  "youtube",
    "v3",
    api_key,
)
```

Listing B.8: Youtube API

Data Format

Each video is returned in the form of a hashmap containing a multitude of information such as publish date, title, description, etc. The appropriate information is extracted from each of these hashmaps. A sample of a youtube JSON video can be found in Appendix C.2.5.

Collection Process

A user's keywords are iterated over and an inner loop of orderings, such as relevance, are provided to the search query. Additionally, if the keyword is in the youtube metadata then the last date of collection is utilized to only include videos that were published after that date. Finally, the user's youtube metadata is updated with the current date on the database. A flowchart of this process can be found in Appendix C.1.8.

B.2 Data Processing

As the data we aim to model is textual in nature, we require first the vectorization of the text and can then apply a machine learning model to it. Text vectorization can include many methods such as bag of words, tfidf, word embeddings, n-grams, etc. If using a transformer model, then they come with their own text vectorizers. The vectorizers used for this project include: tfidf, tfidf-GloVe, bi-grams, and universal sentence encoder. The details of these encoders will be explained in Section 4.3

Appendix C

Data Collection and Samples

C.1 Data Collection of Sources

C.1.1 Fetching User Data

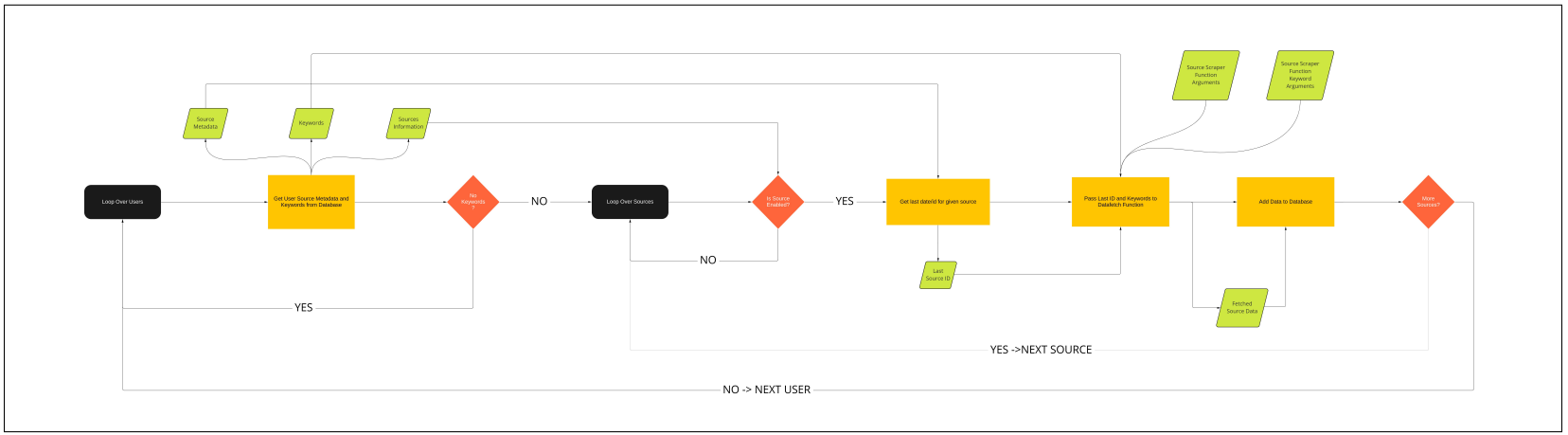


Figure C.1: User Data Fetch Flowchart

C.1.2 Master Fetch Towards Data Science

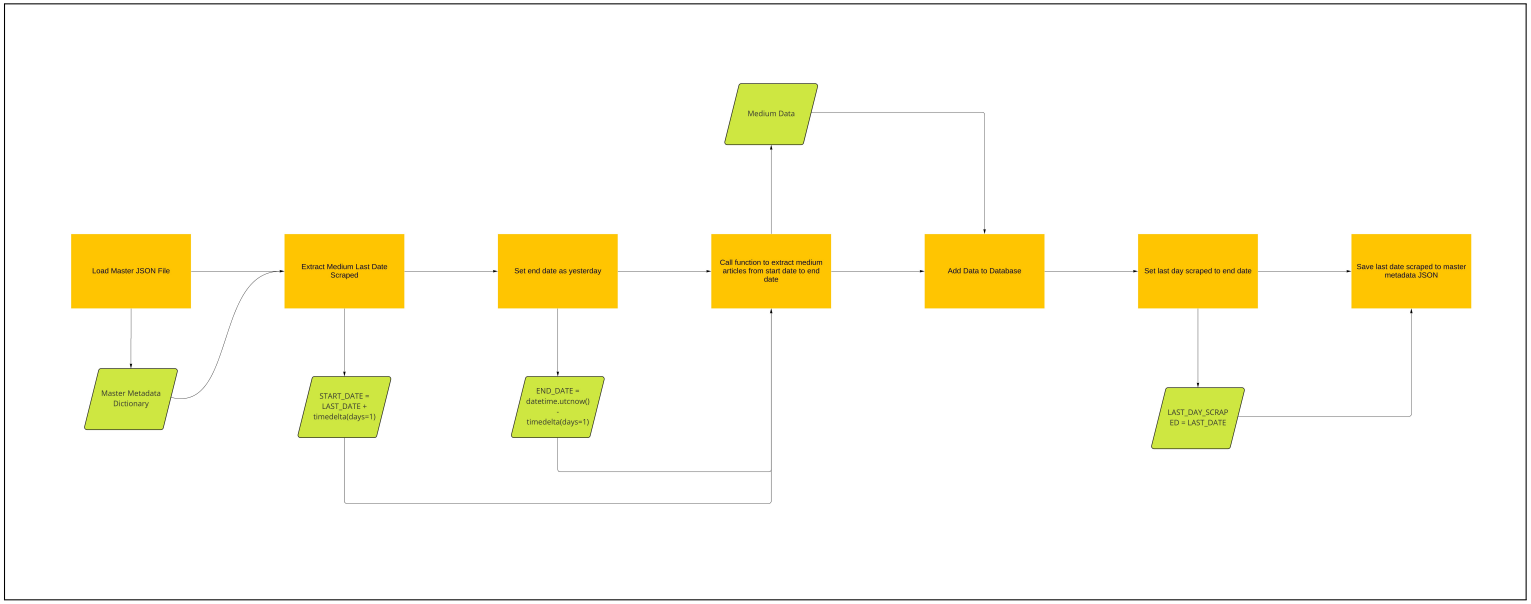


Figure C.2: Towards Data Science - Master Data Fetch Flowchart

C.1.3 Master Fetch PMLR

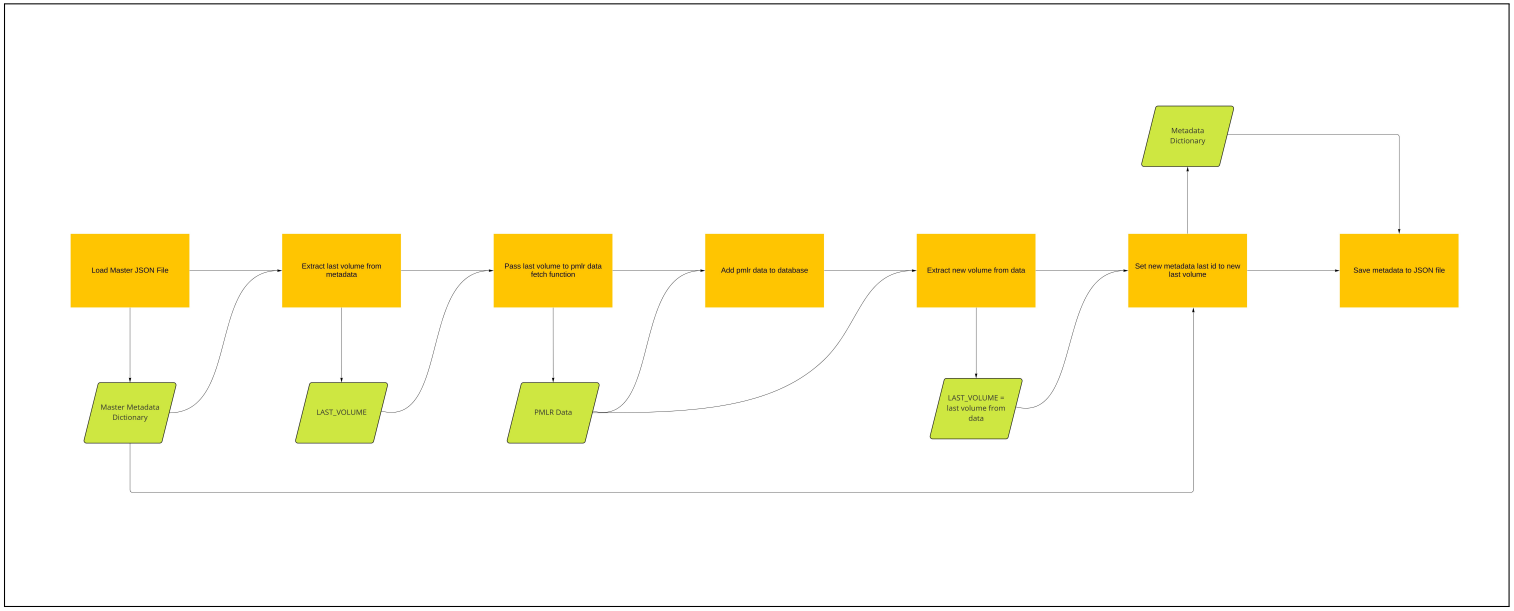


Figure C.3: PMLR - Master Data Fetch Flowchart

C.1.4 Twitter User Data Fetching

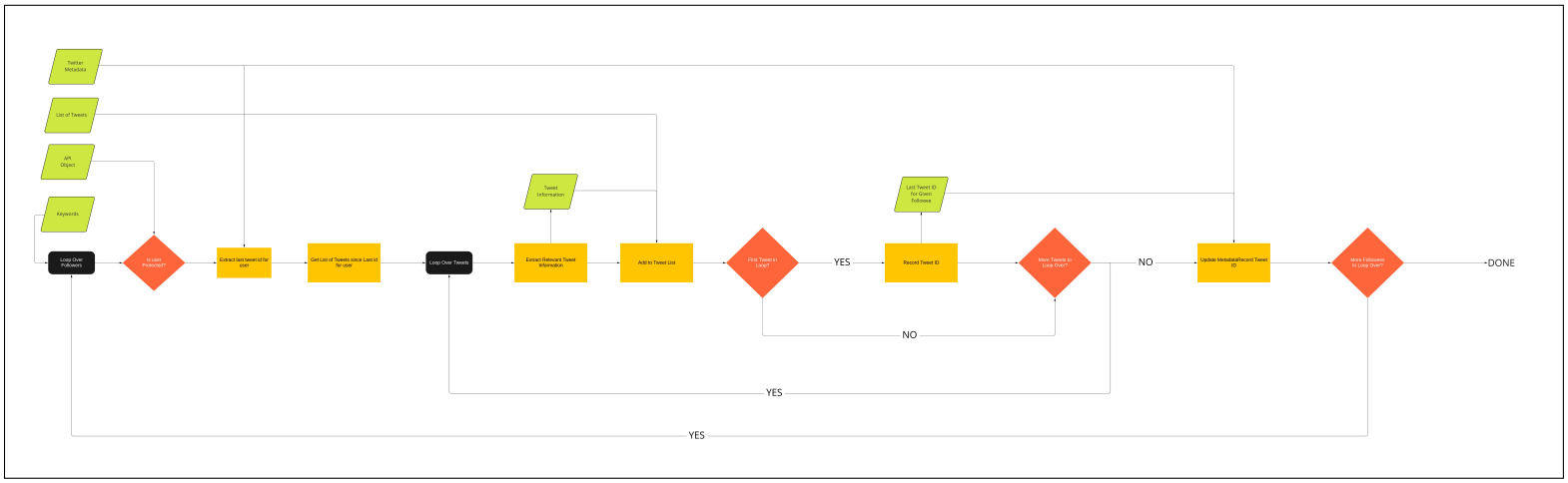


Figure C.4: Twitter Data Fetch Flowchart

C.1.5 Reddit User Data Fetching

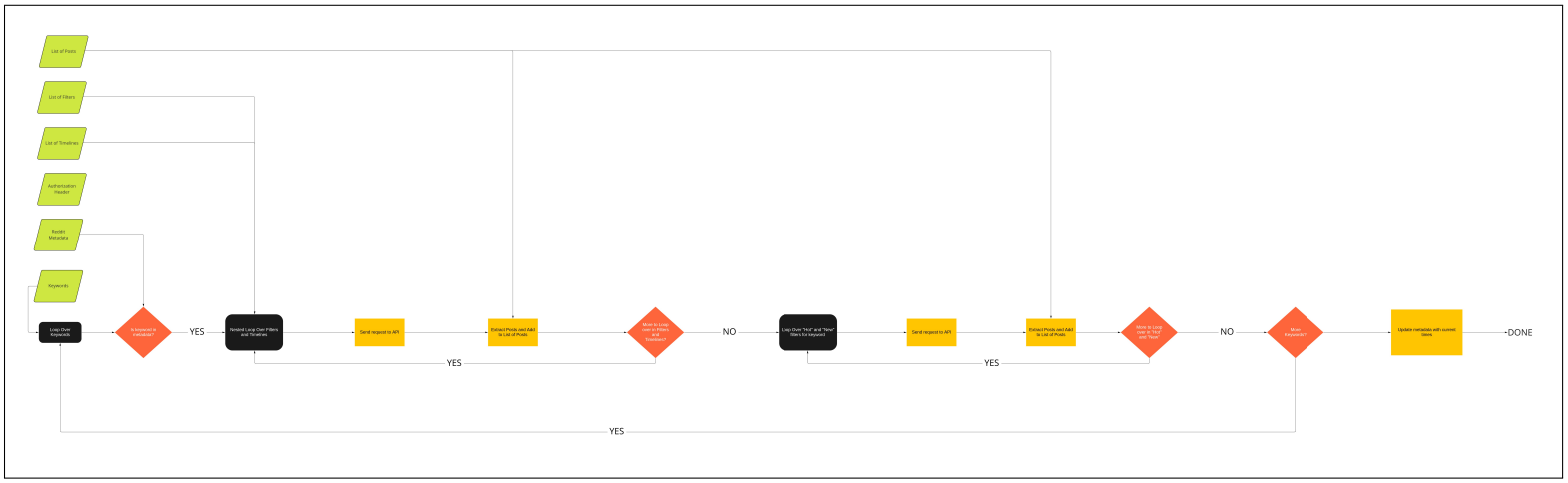


Figure C.5: Reddit Data Fetch Flowchart

C.1.6 PMLR User Data Fetching

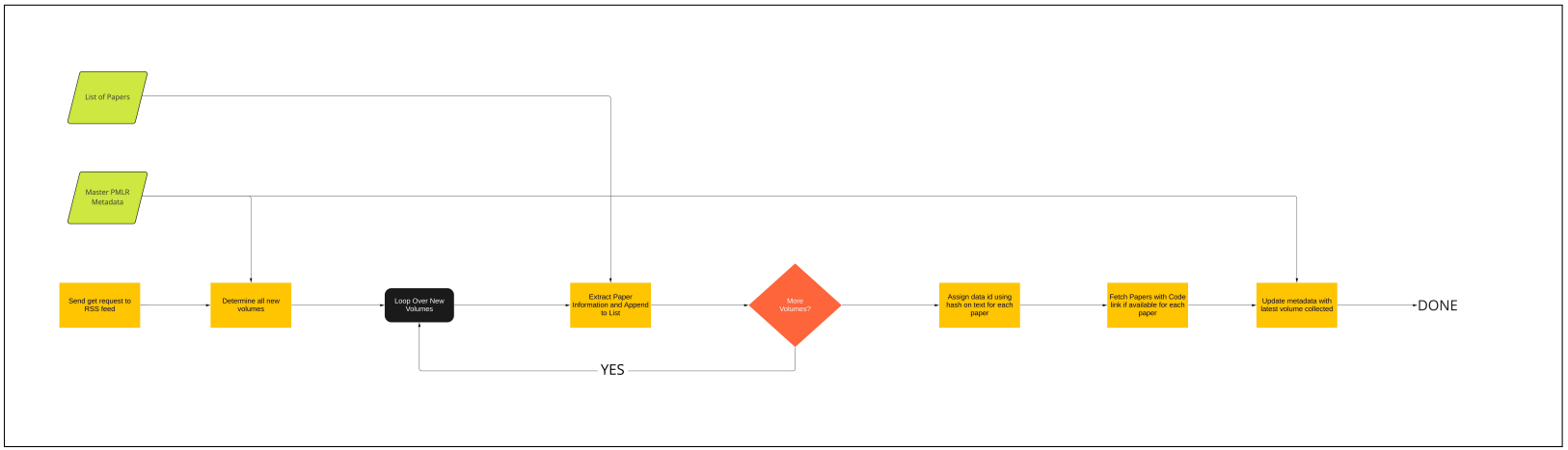


Figure C.6: PMLR Data Fetch Flowchart

C.1.7 Arxiv User Data Fetching

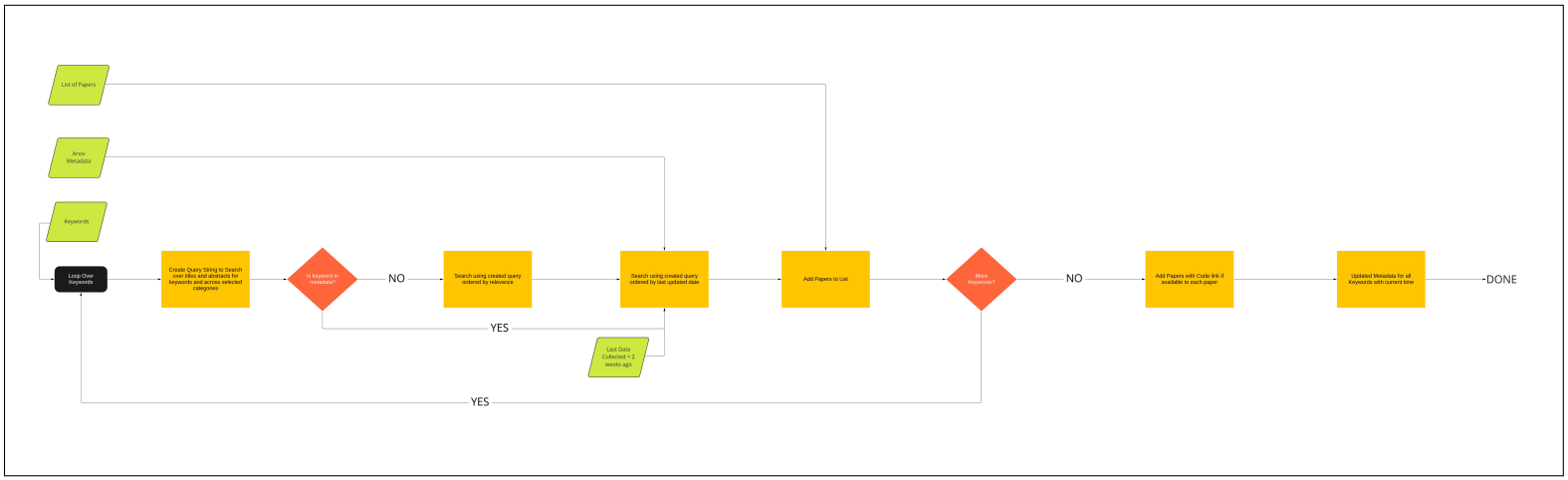


Figure C.7: Arxiv Data Fetch Flowchart

C.1.8 Youtube User Data Fetching

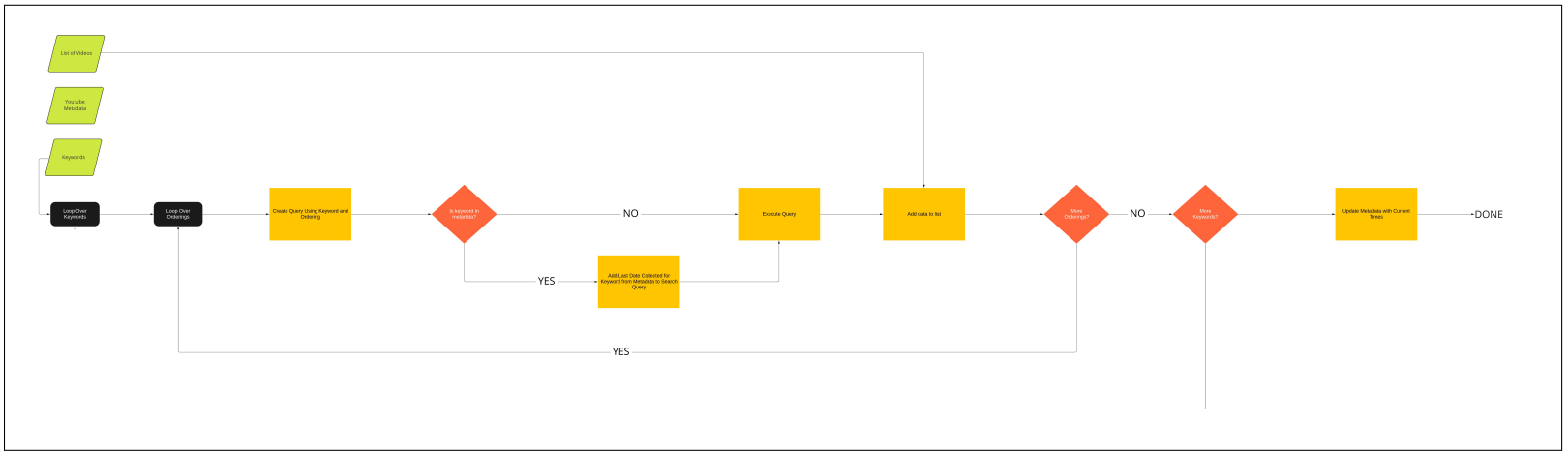


Figure C.8: Youtube Data Fetch Flowchart

C.1.9 Towards Data Science Data Fetching

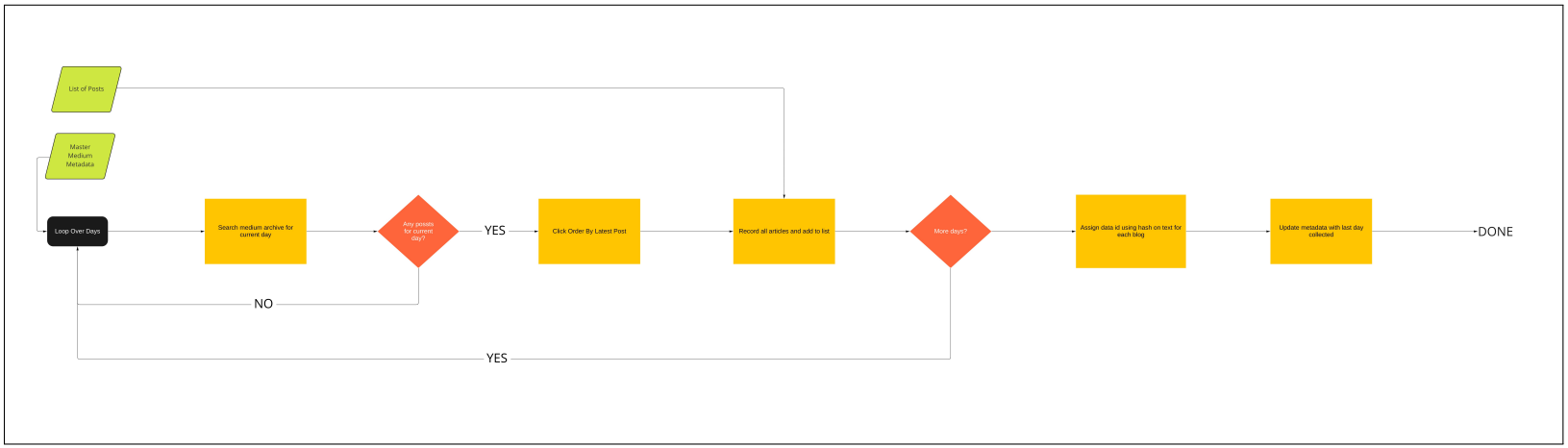


Figure C.9: Towards Data Science Data Fetch Flowchart

C.2 Source Data Samples

C.2.1 Twitter


```
{
   "created_at": "Tue Jun 21 18:31:29 +0000 2022",
   "id": 1539315083871801344,
   "id_str": "1539315083871801344",
   "full_text": "More context about the specific architecture (SIGN) that
    is being used: https://t.co/rZaXlHzShC",
   "truncated": false,
   "display_text_range": [
       0,
       95
   ],
   "entities": {
       "hashtags": [],
       "symbols": [],
       "user_mentions": [],
       "urls": [
           {
               "url": "https://t.co/rZaXlHzShC",
               "expanded_url": "https://mobile.twitter.com/ffabffrasca/
   status/1539257147980120067",
               "display_url": "mobile.twitter.com/ffabffrasca/st\u2026",
               "indices": [
                   72,
                   95
               ]
           }
       ]
   },
   "source": "<a href=\"https://mobile.twitter.com\" rel=\"nofollow\">
   Twitter Web App</a>",
   "in_reply_to_status_id": 1539312867723333633,
   "in_reply_to_status_id_str": "1539312867723333633",
   "in_reply_to_user_id": 50343115,
   "in_reply_to_user_id_str": "50343115",
   "in_reply_to_screen_name": "thomaskipf",
   "user": {
       "id": 50343115,
       "id_str": "50343115",
       "name": "Thomas Kipf",
       "screen_name": "thomaskipf",
```

```
        "location": "Amsterdam",
        "description": "Researcher at Google Brain. @ELLISforEurope Scholar. Deep Learning with Graphs, Abstractions & Objects; e.g. GCNs, Neural Relational Inference, Slot Attention.",
        "url": "https://t.co/NjUOl2tUng",
        "entities": {
            "url": {
                "urls": [
                    {
                        "url": "https://t.co/NjUOl2tUng",
                        "expanded_url": "http://tkipf.github.io",
                        "display_url": "tkipf.github.io",
                        "indices": [
                            0,
                            23
                        ]
                    }
                ]
            },
            "description": {
                "urls": []
            }
        },
        "protected": false,
        "followers_count": 21153,
        "friends_count": 990,
        "listed_count": 349,
        "created_at": "Wed Jun 24 15:54:44 +0000 2009",
        "favourites_count": 5603,
        "utc_offset": null,
        "time_zone": null,
        "geo_enabled": true,
        "verified": false,
        "statuses_count": 1274,
        "lang": null,
        "contributors_enabled": false,
        "is_translator": false,
        "is_translation_enabled": false,
        "profile_background_color": "000000",
        "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
        "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
        "profile_background_tile": false,
```

```
        "profile_image_url": "http://pbs.twimg.com/profile_images
    /778984689986105349/gcfw2w1H_normal.jpg",
        "profile_image_url_https": "https://pbs.twimg.com/profile_images
    /778984689986105349/gcfw2w1H_normal.jpg",
        "profile_banner_url": "https://pbs.twimg.com/profile_banners
    /50343115/1480074597",
        "profile_link_color": "ABB8C2",
        "profile_sidebar_border_color": "000000",
        "profile_sidebar_fill_color": "000000",
        "profile_text_color": "000000",
        "profile_use_background_image": false,
        "has_extended_profile": false,
        "default_profile": false,
        "default_profile_image": false,
        "following": true,
        "follow_request_sent": false,
        "notifications": false,
        "translator_type": "none",
        "withheld_in_countries": []
    },
    "geo": null,
    "coordinates": null,
    "place": null,
    "contributors": null,
    "is_quote_status": true,
    "quoted_status_id": 1539257147980120067,
    "quoted_status_id_str": "1539257147980120067",
    "quoted_status_permalink": {
        "url": "https://t.co/rZaXlHzShC",
        "expanded": "https://mobile.twitter.com/ffabffrasca/status
    /1539257147980120067",
        "display": "mobile.twitter.com/ffabffrasca/st\u2026"
    },
    "quoted_status": {
        "created_at": "Tue Jun 21 14:41:16 +0000 2022",
        "id": 1539257147980120067,
        "id_str": "1539257147980120067",
        "full_text": "The SIGN architecture has been developed at @Twitter
     in our 2020 paper (https://t.co/xkLNLZ6zmm) with @mmbronstein
    @emaros96 @aittalam @b_p_chamberlain @frederickmonti .\n\nSee more in
     this @TwitterEng blogpost: https://t.co/SylhKRcQtO",
        "truncated": false,
        "display_text_range": [
            0,
            233
```

```
        ],
        "entities": {
            "hashtags": [],
            "symbols": [],
            "user_mentions": [
                {
                    "screen_name": "Twitter",
                    "name": "Twitter",
                    "id": 783214,
                    "id_str": "783214",
                    "indices": [
                        44,
                        52
                    ]
                },
                {
                    "screen_name": "mmbronstein",
                    "name": "Michael Bronstein",
                    "id": 62044012,
                    "id_str": "62044012",
                    "indices": [
                        102,
                        114
                    ]
                },
                {
                    "screen_name": "emaros96",
                    "name": "Emanuele Rossi",
                    "id": 4482331636,
                    "id_str": "4482331636",
                    "indices": [
                        115,
                        124
                    ]
                },
                {
                    "screen_name": "aittalam",
                    "name": "Davide Eynard",
                    "id": 14536648,
                    "id_str": "14536648",
                    "indices": [
                        125,
                        134
                    ]
                },
```

```
                {
                    "screen_name": "b_p_chamberlain",
                    "name": "Ben Chamberlain",
                    "id": 25068244,
                    "id_str": "25068244",
                    "indices": [
                        135,
                        151
                    ]
                },
                {
                    "screen_name": "frederickmonti",
                    "name": "Federico Monti",
                    "id": 140733160,
                    "id_str": "140733160",
                    "indices": [
                        152,
                        167
                    ]
                },
                {
                    "screen_name": "TwitterEng",
                    "name": "Twitter Engineering",
                    "id": 6844292,
                    "id_str": "6844292",
                    "indices": [
                        188,
                        199
                    ]
                }
            ],
            "urls": [
                {
                    "url": "https://t.co/xkLNLZ6zmm",
                    "expanded_url": "https://arxiv.org/abs/2004.11198",
                    "display_url": "arxiv.org/abs/2004.11198",
                    "indices": [
                        72,
                        95
                    ]
                },
                {
                    "url": "https://t.co/SylhKRcQtO",
                    "expanded_url": "https://blog.twitter.com/engineering/
    en_us/topics/insights/2021/simple-scalable-graph-neural-networks",
```

```
                    "display_url": "blog.twitter.com/engineering/en\u2026"
,
                    "indices": [
                        210,
                        233
                    ]
                }
            ]
        },
        "source": "<a href=\"https://mobile.twitter.com\" rel=\"nofollow
\">Twitter Web App</a>",
        "in_reply_to_status_id": 1539257145912328197,
        "in_reply_to_status_id_str": "1539257145912328197",
        "in_reply_to_user_id": 1135289016474046464,
        "in_reply_to_user_id_str": "1135289016474046464",
        "in_reply_to_screen_name": "ffabffrasca",
        "user": {
            "id": 1135289016474046464,
            "id_str": "1135289016474046464",
            "name": "Fabrizio Frasca",
            "screen_name": "ffabffrasca",
            "location": "",
            "description": "Researcher @twitter and PhD candidate
@imperialcollege \u2014 Geometric Deep Learning in some of its
various forms \u2014 Previously @fabula_ai and @polimi",
            "url": null,
            "entities": {
                "description": {
                    "urls": []
                }
            },
            "protected": false,
            "followers_count": 844,
            "friends_count": 365,
            "listed_count": 12,
            "created_at": "Sun Jun 02 20:56:09 +0000 2019",
            "favourites_count": 662,
            "utc_offset": null,
            "time_zone": null,
            "geo_enabled": false,
            "verified": false,
            "statuses_count": 235,
            "lang": null,
            "contributors_enabled": false,
            "is_translator": false,
```

```
            "is_translation_enabled": false,
            "profile_background_color": "000000",
            "profile_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.png",
            "profile_background_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png",
            "profile_background_tile": false,
            "profile_image_url": "http://pbs.twimg.com/profile_images/1479256331001511936/2NCnLDwQ_normal.jpg",
            "profile_image_url_https": "https://pbs.twimg.com/profile_images/1479256331001511936/2NCnLDwQ_normal.jpg",
            "profile_banner_url": "https://pbs.twimg.com/profile_banners/1135289016474046464/1641517168",
            "profile_link_color": "0084B4",
            "profile_sidebar_border_color": "000000",
            "profile_sidebar_fill_color": "000000",
            "profile_text_color": "000000",
            "profile_use_background_image": false,
            "has_extended_profile": false,
            "default_profile": false,
            "default_profile_image": false,
            "following": false,
            "follow_request_sent": false,
            "notifications": false,
            "translator_type": "none",
            "withheld_in_countries": []
        },
        "geo": null,
        "coordinates": null,
        "place": null,
        "contributors": null,
        "is_quote_status": false,
        "retweet_count": 5,
        "favorite_count": 20,
        "favorited": false,
        "retweeted": false,
        "possibly_sensitive": false,
        "lang": "en"
    },
    "retweet_count": 0,
    "favorite_count": 6,
    "favorited": false,
    "retweeted": false,
    "possibly_sensitive": false,
    "lang": "en"
```

```
}
```

C.2.2 Reddit


```
    {
   "kind": "t3",
   "data": {
       "approved_at_utc": null,
       "subreddit": "ProgrammerHumor",
       "selftext": "",
       "author_fullname": "t2_dhyv1t4h",
       "saved": false,
       "mod_reason_title": null,
       "gilded": 0,
       "clicked": false,
       "title": "Who wanna learn Machine Learning?",
       "link_flair_richtext": [],
       "subreddit_name_prefixed": "r/ProgrammerHumor",
       "hidden": false,
       "pwls": 6,
       "link_flair_css_class": null,
       "downs": 0,
       "thumbnail_height": 140,
       "top_awarded_type": null,
       "hide_score": false,
       "name": "t3_uu4zgf",
       "quarantine": false,
       "link_flair_text_color": "dark",
       "upvote_ratio": 0.96,
       "author_flair_background_color": null,
       "subreddit_type": "public",
       "ups": 909,
       "total_awards_received": 1,
       "media_embed": {},
       "thumbnail_width": 140,
       "author_flair_template_id": null,
       "is_original_content": false,
       "user_reports": [],
       "secure_media": null,
       "is_reddit_media_domain": true,
       "is_meta": false,
       "category": null,
       "secure_media_embed": {},
       "link_flair_text": null,
```

```
        "can_mod_post": false,
        "score": 909,
        "approved_by": null,
        "is_created_from_ads_ui": false,
        "author_premium": false,
        "thumbnail": "https://b.thumbs.redditmedia.com/SHB64hla9yV9jTpJxW8S4CslNVyGYSOLGzQ6AP1y3Ak.jpg",
        "edited": false,
        "author_flair_css_class": null,
        "author_flair_richtext": [],
        "gildings": {},
        "post_hint": "image",
        "content_categories": null,
        "is_self": false,
        "mod_note": null,
        "created": 1653072841.0,
        "link_flair_type": "text",
        "wls": 6,
        "removed_by_category": null,
        "banned_by": null,
        "author_flair_type": "text",
        "domain": "i.redd.it",
        "allow_live_comments": false,
        "selftext_html": null,
        "likes": null,
        "suggested_sort": null,
        "banned_at_utc": null,
        "url_overridden_by_dest": "https://i.redd.it/24a86vppgo091.jpg",
        "view_count": null,
        "archived": false,
        "no_follow": false,
        "is_crosspostable": true,
        "pinned": false,
        "over_18": false,
        "preview": {
            "images": [
                {
                    "source": {
                        "url": "https://preview.redd.it/24a86vppgo091.jpg?auto=webp&amp;s=62de89b51525df3fb9cda80d62a45198e40a5edb",
                        "width": 500,
                        "height": 672
                    },
                    "resolutions": [
                        {
```

```
                            "url": "https://preview.redd.it/24a86vppgo091.jpg?width=108&amp;crop=smart&amp;auto=webp&amp;s=27714116dcf9380385286d7ce29f06ed3caa6a48",
                            "width": 108,
                            "height": 145
                        },
                        {
                            "url": "https://preview.redd.it/24a86vppgo091.jpg?width=216&amp;crop=smart&amp;auto=webp&amp;s=ffe07d83cbca38a65d60980d98a97f2b6b94e382",
                            "width": 216,
                            "height": 290
                        },
                        {
                            "url": "https://preview.redd.it/24a86vppgo091.jpg?width=320&amp;crop=smart&amp;auto=webp&amp;s=f129adf7c9d2134687a4f226b892e8e59fc6c189",
                            "width": 320,
                            "height": 430
                        }
                    ],
                    "variants": {},
                    "id": "8dKKcC4j8f6_7cOGLg2vTQXiEJqy48Z_PT_1goGSRX0"
                }
            ],
            "enabled": true
        },
        "all_awardings": [
            {
                "giver_coin_reward": null,
                "subreddit_id": null,
                "is_new": false,
                "days_of_drip_extension": null,
                "coin_price": 50,
                "id": "award_02d9ab2c-162e-4c01-8438-317a016ed3d9",
                "penny_donate": null,
                "award_sub_type": "GLOBAL",
                "coin_reward": 0,
                "icon_url": "https://i.redd.it/award_images/t5_q0gj4/p4yzxkaed5f61_oldtakemyenergy.png",
                "days_of_premium": null,
                "tiers_by_required_awardings": null,
                "resized_icons": [
                    {
                        "url": "https://preview.redd.it/award_images/
```

```
t5_q0gj4/p4yzxkaed5f61_oldtakemyenergy.png?width=16&amp;height=16&amp;auto=webp&amp;s=10034f3fdf8214c8377134bb60c5b832d4bbf588",
                    "width": 16,
                    "height": 16
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/p4yzxkaed5f61_oldtakemyenergy.png?width=32&amp;height=32&amp;auto=webp&amp;s=100f785bf261fa9452a5d82ee0ef0793369dbfa5",
                    "width": 32,
                    "height": 32
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/p4yzxkaed5f61_oldtakemyenergy.png?width=48&amp;height=48&amp;auto=webp&amp;s=b15d030fdfbbe4af4a5b34ab9dc90a174df40a23",
                    "width": 48,
                    "height": 48
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/p4yzxkaed5f61_oldtakemyenergy.png?width=64&amp;height=64&amp;auto=webp&amp;s=601c75be6ee30dc4b47a5c65d64dea9a185502a1",
                    "width": 64,
                    "height": 64
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/p4yzxkaed5f61_oldtakemyenergy.png?width=128&amp;height=128&amp;auto=webp&amp;s=540f36e65c0e2f1347fe32020e4a1565e3680437",
                    "width": 128,
                    "height": 128
                }
            ],
            "icon_width": 2048,
            "static_icon_width": 2048,
            "start_date": null,
            "is_enabled": true,
            "awardings_required_to_grant_benefits": null,
            "description": "I'm in this with you.",
            "end_date": null,
            "sticky_duration_seconds": null,
            "subreddit_coin_reward": 0,
            "count": 1,
            "static_icon_height": 2048,
```

```
            "name": "Take My Energy",
            "resized_static_icons": [
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/jtw7x06j68361_TakeMyEnergyElf.png?width=16&amp;height=16&amp;auto=webp&amp;s=045db73f47a9513c44823d132b4c393ab9241b6a",
                    "width": 16,
                    "height": 16
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/jtw7x06j68361_TakeMyEnergyElf.png?width=32&amp;height=32&amp;auto=webp&amp;s=298a02e0edbb5b5e293087eeede63802cbe1d2c7",
                    "width": 32,
                    "height": 32
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/jtw7x06j68361_TakeMyEnergyElf.png?width=48&amp;height=48&amp;auto=webp&amp;s=7d06d606eb23dbcd6dbe39ee0e60588c5eb89065",
                    "width": 48,
                    "height": 48
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/jtw7x06j68361_TakeMyEnergyElf.png?width=64&amp;height=64&amp;auto=webp&amp;s=ecd9854b14104a36a210028c43420f0dababd96b",
                    "width": 64,
                    "height": 64
                },
                {
                    "url": "https://preview.redd.it/award_images/t5_q0gj4/jtw7x06j68361_TakeMyEnergyElf.png?width=128&amp;height=128&amp;auto=webp&amp;s=0d5d7b92c1d66aff435f2ad32e6330ca2b971f6d",
                    "width": 128,
                    "height": 128
                }
            ],
            "icon_format": "PNG",
            "icon_height": 2048,
            "penny_price": 0,
            "award_type": "global",
            "static_icon_url": "https://i.redd.it/award_images/t5_q0gj4/jtw7x06j68361_TakeMyEnergyElf.png"
        }
```

```
        ],
        "awarders": [],
        "media_only": false,
        "can_gild": true,
        "spoiler": false,
        "locked": false,
        "author_flair_text": null,
        "treatment_tags": [],
        "visited": false,
        "removed_by": null,
        "num_reports": null,
        "distinguished": null,
        "subreddit_id": "t5_2tex6",
        "author_is_blocked": false,
        "mod_reason_by": null,
        "removal_reason": null,
        "link_flair_background_color": "",
        "id": "uu4zgf",
        "is_robot_indexable": true,
        "report_reasons": null,
        "author": "SuPythony",
        "discussion_type": null,
        "num_comments": 37,
        "send_replies": true,
        "whitelist_status": "all_ads",
        "contest_mode": false,
        "mod_reports": [],
        "author_patreon_flair": false,
        "author_flair_text_color": null,
        "permalink": "/r/ProgrammerHumor/comments/uu4zgf/
    who_wanna_learn_machine_learning/",
        "parent_whitelist_status": "all_ads",
        "stickied": false,
        "url": "https://i.redd.it/24a86vppgo091.jpg",
        "subreddit_subscribers": 2175627,
        "created_utc": 1653072841.0,
        "num_crossposts": 0,
        "media": null,
        "is_video": false
    }
}
```


C.2.3 PMLR

```
    <bound method Tag.prettify of <?xml version="1.0" encoding="utf-8"?>
<rss version="2.0" xmlns:atom="http://www.w3.org/2005/Atom">
<channel>
<title>Proceedings of Machine Learning Research</title>
<description>The Proceedings of Machine Learning Research (formerly JMLR
   Workshop and Conference Proceedings) is a series aimed  specifically
   at publishing machine learning research presented at workshops and
   conferences. Each volume is separately  titled and associated with a
   particular workshop or conference. Volumes are published online on
   the PMLR web site.
The Series Editors are Neil D. Lawrence and Mark Reid.
</description>
<link>https://proceedings.mlr.press/</link>
<atom:link href="https://proceedings.mlr.press/feed.xml" rel="self" type=
   "application/rss+xml"/>
<pubDate>Wed, 20 Jul 2022 19:16:03 +0000</pubDate>
<lastBuildDate>Wed, 20 Jul 2022 19:16:03 +0000</lastBuildDate>
<generator>Jekyll v3.9.2</generator>
<item>
<title>v1</title>
<description>Gaussian Processes in Practice</description>
<link>https://proceedings.mlr.press/v1</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v1</guid>
</item>
<item>
<title>v2</title>
<description>AISTATS 2007 Proceedings</description>
<link>https://proceedings.mlr.press/v2</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v2</guid>
</item>
<item>
<title>v3</title>
<description>Causation and Prediction Challenge (WCCI 2008)</description>
<link>https://proceedings.mlr.press/v3</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v3</guid>
</item>
<item>
<title>v4</title>
<description>New Challenges for Feature Selection in Data Mining and
   Knowledge Discovery</description>
<link>https://proceedings.mlr.press/v4</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v4</guid>
</item>
<item>
<title>v5</title>
```

```
<description>AISTATS 2009 Proceedings</description>
<link>https://proceedings.mlr.press/v5</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v5</guid>
</item>
<item>
<title>v6</title>
<description>Causality: Objectives and Assessment (NIPS 2008 Workshop)</description>
<link>https://proceedings.mlr.press/v6</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v6</guid>
</item>
<item>
<title>v7</title>
<description>Proceedings of KDD-Cup 2009 competition</description>
<link>https://proceedings.mlr.press/v7</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v7</guid>
</item>
<item>
<title>v8</title>
<description>Proceedings of MLSB 2009</description>
<link>https://proceedings.mlr.press/v8</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v8</guid>
</item>
<item>
<title>v9</title>
<description>AISTATS 2010 Proceedings</description>
<link>https://proceedings.mlr.press/v9</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v9</guid>
</item>
<item>
<title>v10</title>
<description>FSDM 2010 Proceedings</description>
<link>https://proceedings.mlr.press/v10</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v10</guid>
</item>
<item>
<title>v11</title>
<description>Workshop on Applications of Pattern Analysis (WAPA) 2010 Proceedings</description>
<link>https://proceedings.mlr.press/v11</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v11</guid>
</item>
<item>
<title>v12</title>
<description>NIPS Mini-Symposium on Causality in Time Series</description
```

```
   >
<link>https://proceedings.mlr.press/v12</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v12</guid>
</item>
<item>
<title>v13</title>
<description>ACML 2010 Proceedings</description>
<link>https://proceedings.mlr.press/v13</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v13</guid>
</item>
<item>
<title>v14</title>
<description>Proceedings of the Learning to Rank Challenge</description>
<link>https://proceedings.mlr.press/v14</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v14</guid>
</item>
<item>
<title>v15</title>
<description>AISTATS 2011 Proceedings</description>
<link>https://proceedings.mlr.press/v15</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v15</guid>
</item>
<item>
<title>v16</title>
<description>Active Learning and Experimental Design Workshop</
   description>
<link>https://proceedings.mlr.press/v16</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v16</guid>
</item>
<item>
<title>v17</title>
<description>Workshop on Applications of Pattern Analysis (WAPA) 2011
   Proceedings</description>
<link>https://proceedings.mlr.press/v17</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v17</guid>
</item>
<item>
<title>v18</title>
<description>KDD Cup 2011</description>
<link>https://proceedings.mlr.press/v18</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v18</guid>
</item>
<item>
<title>v19</title>
<description>COLT 2011 Proceedings</description>
```

```
<link>https://proceedings.mlr.press/v19</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v19</guid>
</item>
<item>
<title>v20</title>
<description>ACML 2011 Proceedings</description>
<link>https://proceedings.mlr.press/v20</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v20</guid>
</item>
<item>
<title>v21</title>
<description>ICGI 2012 Proceedings</description>
<link>https://proceedings.mlr.press/v21</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v21</guid>
</item>
<item>
<title>v22</title>
<description>AISTATS 2012 Proceedings</description>
<link>https://proceedings.mlr.press/v22</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v22</guid>
</item>
<item>
<title>v23</title>
<description>COLT 2012 Proceedings</description>
<link>https://proceedings.mlr.press/v23</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v23</guid>
</item>
<item>
<title>v24</title>
<description>EWRL 2012 Proceedings</description>
<link>https://proceedings.mlr.press/v24</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v24</guid>
</item>
<item>
<title>v25</title>
<description>ACML 2012 Proceedings</description>
<link>https://proceedings.mlr.press/v25</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v25</guid>
</item>
<item>
<title>v26</title>
<description>On-line Trading of Exploration and Exploitation 2011
   Proceedings</description>
<link>https://proceedings.mlr.press/v26</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v26</guid>
```

```
</item>
<item>
<title>v27</title>
<description>Unsupervised and Transfer Learning Workshop at ICML 2011</
   description>
<link>https://proceedings.mlr.press/v27</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v27</guid>
</item>
<item>
<title>v28</title>
<description>ICML 2013 Proceedings</description>
<link>https://proceedings.mlr.press/v28</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v28</guid>
</item>
<item>
<title>v29</title>
<description>ACML 2013 Proceedings</description>
<link>https://proceedings.mlr.press/v29</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v29</guid>
</item>
<item>
<title>v30</title>
<description>COLT 2013 Proceedings</description>
<link>https://proceedings.mlr.press/v30</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v30</guid>
</item>
<item>
<title>v31</title>
<description>AISTATS 2013 Proceedings</description>
<link>https://proceedings.mlr.press/v31</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v31</guid>
</item>
<item>
<title>v32</title>
<description>ICML 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v32</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v32</guid>
</item>
<item>
<title>v33</title>
<description>AISTATS 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v33</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v33</guid>
</item>
<item>
```

```
<title>v34</title>
<description>ICGI 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v34</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v34</guid>
</item>
<item>
<title>v35</title>
<description>COLT 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v35</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v35</guid>
</item>
<item>
<title>v36</title>
<description>BIGMINE 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v36</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v36</guid>
</item>
<item>
<title>v37</title>
<description>ICML 2015 Proceedings</description>
<link>https://proceedings.mlr.press/v37</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v37</guid>
</item>
<item>
<title>v38</title>
<description>AISTATS 2015 Proceedings</description>
<link>https://proceedings.mlr.press/v38</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v38</guid>
</item>
<item>
<title>v39</title>
<description>ACML 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v39</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v39</guid>
</item>
<item>
<title>v40</title>
<description>COLT 2015 Proceedings</description>
<link>https://proceedings.mlr.press/v40</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v40</guid>
</item>
<item>
<title>v41</title>
<description>BigMine 2015 Proceedings</description>
<link>https://proceedings.mlr.press/v41</link>
```

```
<guid isPermaLink="true">https://proceedings.mlr.press/v41</guid>
</item>
<item>
<title>v42</title>
<description>HEPML 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v42</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v42</guid>
</item>
<item>
<title>v43</title>
<description>MLIS 2015 Proceedings</description>
<link>https://proceedings.mlr.press/v43</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v43</guid>
</item>
<item>
<title>v44</title>
<description>NIPS Workshop on Feature Extraction 2015 Proceedings</
   description>
<link>https://proceedings.mlr.press/v44</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v44</guid>
</item>
<item>
<title>v45</title>
<description>ACML 2015 Proceedings</description>
<link>https://proceedings.mlr.press/v45</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v45</guid>
</item>
<item>
<title>v46</title>
<description>Neural Connectomics Workshop 2014 Proceedings</description>
<link>https://proceedings.mlr.press/v46</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v46</guid>
</item>
<item>
<title>v47</title>
<description>ECML/PKDD Workshop on Statistically Sound Data Mining 2014
   Proceedings</description>
<link>https://proceedings.mlr.press/v47</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v47</guid>
</item>
<item>
<title>v48</title>
<description>ICML 2016 Proceedings</description>
<link>https://proceedings.mlr.press/v48</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v48</guid>
```

```
</item>
<item>
<title>v49</title>
<description>COLT 2016 Proceedings</description>
<link>https://proceedings.mlr.press/v49</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v49</guid>
</item>
<item>
<title>v50</title>
<description>PAPIs 2015 Proceedings</description>
<link>https://proceedings.mlr.press/v50</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v50</guid>
</item>
<item>
<title>v51</title>
<description>AISTATS 2016 Proceedings</description>
<link>https://proceedings.mlr.press/v51</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v51</guid>
</item>
<item>
<title>v52</title>
<description>PGM 2016 Proceedings</description>
<link>https://proceedings.mlr.press/v52</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v52</guid>
</item>
<item>
<title>v53</title>
<description>BigMine 2016 Proceedings</description>
<link>https://proceedings.mlr.press/v53</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v53</guid>
</item>
<item>
<title>v55</title>
<description>NIPS 2016 Time Series Workshop</description>
<link>https://proceedings.mlr.press/v55</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v55</guid>
</item>
<item>
<title>v56</title>
<description>Proceedings of Machine Learning for Healthcare 2016</
   description>
<link>https://proceedings.mlr.press/v56</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v56</guid>
</item>
<item>
```

```
<title>v57</title>
<description>ICGI 2016 Proceedings</description>
<link>https://proceedings.mlr.press/v57</link>
<guid isPermaLink="true">https://proceedings.mlr.press/v57</guid>
</item>
</channel>
</rss>>
```

C.2.4 Arxiv

```
 {
"bozo": false,
"entries": [
    {
        "id": "http://arxiv.org/abs/cond-mat/0102536v1",
        "guidislink": true,
        "link": "http://arxiv.org/abs/cond-mat/0102536v1",
        "updated": "2001-02-28T20:12:09Z",
        "updated_parsed": [
            2001,
            2,
            28,
            20,
            12,
            9,
            2,
            59,
            0
        ],
        "published": "2001-02-28T20:12:09Z",
        "published_parsed": [
            2001,
            2,
            28,
            20,
            12,
            9,
            2,
            59,
            0
        ],
        "title": "Impact of Electron-Electron Cusp on Configuration Interaction Energies",
        "title_detail": {
```

```
            "type": "text/plain",
            "language": null,
            "base": "",
            "value": "Impact of Electron-Electron Cusp on
Configuration Interaction Energies"
        },
        "summary": "The effect of the electron-electron cusp on the
convergence of configuration\ninteraction (CI) wave functions is
examined. By analogy with the\npseudopotential approach for electron-
ion interactions, an effective\nelectron-electron interaction is
developed which closely reproduces the\nscattering of the Coulomb
interaction but is smooth and finite at zero\nelectron-electron
separation. The exact many-electron wave function for this\nsmooth
effective interaction has no cusp at zero electron-electron
separation.\nWe perform CI and quantum Monte Carlo calculations for
He and Be atoms, both\nwith the Coulomb electron-electron interaction
 and with the smooth effective\nelectron-electron interaction. We
find that convergence of the CI expansion of\nthe wave function for
the smooth electron-electron interaction is not\nsignificantly
improved compared with that for the divergent Coulomb interaction\
nfor energy differences on the order of 1 mHartree. This shows that,
contrary to\npopular belief, description of the electron-electron
cusp is not a limiting\nfactor, to within chemical accuracy, for CI
calculations.",
        "summary_detail": {
            "type": "text/plain",
            "language": null,
            "base": "",
            "value": "The effect of the electron-electron cusp on the
convergence of configuration\ninteraction (CI) wave functions is
examined. By analogy with the\npseudopotential approach for electron-
ion interactions, an effective\nelectron-electron interaction is
developed which closely reproduces the\nscattering of the Coulomb
interaction but is smooth and finite at zero\nelectron-electron
separation. The exact many-electron wave function for this\nsmooth
effective interaction has no cusp at zero electron-electron
separation.\nWe perform CI and quantum Monte Carlo calculations for
He and Be atoms, both\nwith the Coulomb electron-electron interaction
 and with the smooth effective\nelectron-electron interaction. We
find that convergence of the CI expansion of\nthe wave function for
the smooth electron-electron interaction is not\nsignificantly
improved compared with that for the divergent Coulomb interaction\
nfor energy differences on the order of 1 mHartree. This shows that,
contrary to\npopular belief, description of the electron-electron
cusp is not a limiting\nfactor, to within chemical accuracy, for CI
```

```
    calculations."
        },
        "authors": [
            {
                "name": "David Prendergast"
            },
            {
                "name": "M. Nolan"
            },
            {
                "name": "Claudia Filippi"
            },
            {
                "name": "Stephen Fahy"
            },
            {
                "name": "J. C. Greer"
            }
        ],
        "author_detail": {
            "name": "J. C. Greer"
        },
        "arxiv_affiliation": "NMRC, University College, Cork, Ireland"
    ,
        "author": "J. C. Greer",
        "arxiv_doi": "10.1063/1.1383585",
        "links": [
            {
                "title": "doi",
                "href": "http://dx.doi.org/10.1063/1.1383585",
                "rel": "related",
                "type": "text/html"
            },
            {
                "href": "http://arxiv.org/abs/cond-mat/0102536v1",
                "rel": "alternate",
                "type": "text/html"
            },
            {
                "title": "pdf",
                "href": "http://arxiv.org/pdf/cond-mat/0102536v1",
                "rel": "related",
                "type": "application/pdf"
            }
        ],
```

```
            "arxiv_comment": "11 pages, 6 figures, 3 tables, LaTeX209,
    submitted to The Journal of\n  Chemical Physics",
            "arxiv_journal_ref": "J. Chem. Phys. 115, 1626 (2001)",
            "arxiv_primary_category": {
                "term": "cond-mat.str-el",
                "scheme": "http://arxiv.org/schemas/atom"
            },
            "tags": [
                {
                    "term": "cond-mat.str-el",
                    "scheme": "http://arxiv.org/schemas/atom",
                    "label": null
                }
            ]
        }
    ],
    "feed": {
        "links": [
            {
                "href": "http://arxiv.org/api/query?search_query%3Dall%3
    Aelectron%26id_list%3D%26start%3D0%26max_results%3D1",
                "rel": "self",
                "type": "application/atom+xml"
            }
        ],
        "title": "ArXiv Query: search_query=all:electron&amp;id_list=&amp;
    start=0&amp;max_results=1",
        "title_detail": {
            "type": "text/html",
            "language": null,
            "base": "",
            "value": "ArXiv Query: search_query=all:electron&amp;id_list=&
    amp;start=0&amp;max_results=1"
        },
        "id": "http://arxiv.org/api/cHxbiOdZaP56ODnBPIenZhzg5f8",
        "guidislink": true,
        "link": "http://arxiv.org/api/cHxbiOdZaP56ODnBPIenZhzg5f8",
        "updated": "2022-07-28T00:00:00-04:00",
        "updated_parsed": [
            2022,
            7,
            28,
            4,
            0,
            0,
```

```
            3,
            209,
            0
        ],
        "opensearch_totalresults": "186067",
        "opensearch_startindex": "0",
        "opensearch_itemsperpage": "1"
    },
    "headers": {},
    "encoding": "utf-8",
    "version": "atom10",
    "namespaces": {
        "": "http://www.w3.org/2005/Atom",
        "opensearch": "http://a9.com/-/spec/opensearch/1.1/",
        "arxiv": "http://arxiv.org/schemas/atom"
    }
}
```


C.2.5 Youtube


```
    {
    "kind": "youtube#searchResult",
    "etag": "p-6m2g1BvyeUpHjRm5e_-B_eBIA",
    "id": {
        "kind": "youtube#video",
        "videoId": "ukzFI9rgwfU"
    },
    "snippet": {
        "publishedAt": "2018-09-19T14:57:02Z",
        "channelId": "UCsvqVGtbbyHaMoevxPAq9Fg",
        "title": "Machine Learning Basics | What Is Machine Learning? | Introduction To Machine Learning | Simplilearn",
        "description": "Enroll for FREE Machine Learning Course & Get your Completion Certificate: ...",
        "thumbnails": {
            "default": {
                "url": "https://i.ytimg.com/vi/ukzFI9rgwfU/default.jpg",
                "width": 120,
                "height": 90
            },
            "medium": {
                "url": "https://i.ytimg.com/vi/ukzFI9rgwfU/mqdefault.jpg",
                "width": 320,
                "height": 180
```

```
            },
            "high": {
                "url": "https://i.ytimg.com/vi/ukzFI9rgwfU/hqdefault.jpg",
                "width": 480,
                "height": 360
            }
        },
        "channelTitle": "Simplilearn",
        "liveBroadcastContent": "none",
        "publishTime": "2018-09-19T14:57:02Z"
    }
}
```

C.2.6 Towards Dada Science

```
    <!DOCTYPE html>
<head>
   <link rel="stylesheet" href="https://cdn-static-1.medium.com/_/fp/css/
   main-branding-base.e1-R_XYAO1Dxl4Gmojo8tw.css">
</head>
<body>
           <div class="streamItem streamItem--postPreview js-streamItem">
               <div class="cardChromeless u-marginTop20 u-paddingTop10 u-
   paddingBottom15 u-paddingLeft20 u-paddingRight20">
                   <div class="postArticle postArticle--short js-
   postArticle js-trackPostPresentation js-trackPostScrolls" data-post-
   id="ede4318cc700" data-source="---------2----------------------"><
   div class="u-clearfix u-marginBottom15 u-paddingTop5"><div class="
   postMetaInline u-floatLeft u-sm-maxWidthFullWidth"><div class="u-
   flexCenter"><div class="postMetaInline-avatar u-flex0"><a class="link
    u-baseColor--link avatar" data-action="show-user-card" data-action-
   type="hover" data-action-value="563d09da62a" data-collection-slug="
   towards-data-science" data-user-id="563d09da62a" dir="auto" href="
   https://towardsdatascience.com/@bindichen"><div class="u-relative u-
   inlineBlock u-flex0"><img alt="Go to the profile of B. Chen" class="
   avatar-image u-size36x36 u-xs-size32x32" src="https://cdn-images-1.
   medium.com/fit/c/36/36/2*UyCeCsiOgZf8pipPJIrFng.png"><div class="
   avatar-halo u-absolute u-textColorGreenNormal svgIcon" style="
   display:flex; flex-direction:row; align-items: center; justify-
   content: center; height:calc(100% + 25px); width:calc(100% + 25px);
   pointer-events: none; top:50%; left:50%; transform:translateY(-50%)
   translateX(-50%); "><svg height="39" style="width: calc(100% - 18px);
    height: calc(100% - 18px);" viewBox="0 0 39 39" width="39" xmlns="
   http://www.w3.org/2000/svg"><path clip-rule="evenodd" d="M19.5 1C11
```

```
.8339 1 5.168 5.74599 1.90845 12.7115L1 12.2885C4.40345 5.01551
11.3974 0 19.5 0C27.6026 0 34.5966 5.01551 38 12.2885L37.0915 12.7115
C33.832 5.74599 27.1661 1 19.5 1ZM1.90845 26.2885C5.168 33.254
11.8339 38 19.5 38C27.1661 38 33.832 33.254 37.0915 26.2885L38
26.7115C34.5966 33.9845 27.6026 39 19.5 39C11.3974 39 4.40345 33.9845
 1 26.7115L1.90845 26.2885Z" fill-rule="evenodd"></path></svg></div><
/div></a></div><div class="postMetaInline postMetaInline-authorLockup
 ui-captionStrong u-flex1 u-noWrapWithEllipsis"><a class="ds-link ds-
link--styleSubtle link link--darken link--accent u-accentColor--
textNormal u-accentColor--textDarken" data-action="show-user-card"
data-action-source="---------2----------------------" data-action-
type="hover" data-action-value="563d09da62a" data-collection-slug="
towards-data-science" data-user-id="563d09da62a" dir="auto" href="
https://towardsdatascience.com/@bindichen?source
=---------2----------------------">B. Chen</a> in <a class="ds-link
ds-link--styleSubtle link--darken link--accent u-accentColor--
textNormal" data-action="show-collection-card" data-action-source="
---------2----------------------" data-action-type="hover" data-
action-value="towards-data-science" data-collection-slug="towards-
data-science" href="https://towardsdatascience.com?source
=---------2----------------------">Towards Data Science</a><div
class="ui-caption u-fontSize12 u-baseColor--textNormal u-
textColorNormal js-postMetaInlineSupplemental"><a class="link link--
darken" data-action="open-post" data-action-source="preview-listing"
data-action-value="https://towardsdatascience.com/5-tricks-to-master-
pandas-append-ede4318cc700?source=---------2----------------------"
href="https://towardsdatascience.com/5-tricks-to-master-pandas-append
-ede4318cc700?source=---------2----------------------"><time
datetime="2022-01-17T23:17:35.549Z">Jan 17</time></a><span class="
middotDivider u-fontSize12"></span><span class="readingTime" title="6
 min read"></span><span class="u-paddingLeft4"><span class="svgIcon
svgIcon--star svgIcon--15px"><svg class="svgIcon-use" height="15"
width="15"><path d="M7.438 2.324c.034-.099.09-.099.123 0l1.2 3.53a
.29.29 0 00.26.19h3.884c.11 0 .127.049.038.111L9.8 8.327a.271.271 0
00-.099.291l1.2 3.53c.034.1-.011.131-.098.069l-3.142-2.18a.303.303 0
00-.32 0l-3.145 2.182c-.087.06-.132.03-.099-.068l1.2-3.53a.271.271 0
00-.098-.292L2.056 6.146c-.087-.06-.071-.112.038-.112h3.884a.29.29 0
00.26-.19l1.2-3.52z"></path></svg></span></span></div></div></div></
div></div><div><a class="" data-action-source="
---------2----------------------" href="https://towardsdatascience.
com/5-tricks-to-master-pandas-append-ede4318cc700?source
=---------2----------------------"><div class="postArticle-content
js-postField"><section class="section section--body section--first
section--last"><div class="section-divider"><hr class="section-
divider"></div><div class="section-content"><div class="section-inner
```

```
 sectionLayout--insetColumn"><figure class="graf graf--figure graf--
layoutCroppedHeightPreview graf--leading" id="previewImage" name="
previewImage"><div class="aspectRatioPlaceholder is-locked"><div
class="aspectRatioPlaceholder-fill" style="padding-bottom: 30%;"></
div><div class="progressiveMedia js-progressiveMedia graf-image is-
canvasLoaded" data-height="2910" data-image-id="1*
zGHaOHYGV4LRCqSlyAtMBg.jpeg" data-is-featured="true" data-scroll="
native" data-width="4366"><img class="progressiveMedia-thumbnail js-
progressiveMedia-thumbnail" crossorigin="anonymous" src="https://cdn-
images-1.medium.com/freeze/fit/t/30/9/1*zGHaOHYGV4LRCqSlyAtMBg.jpeg?q
=20"><canvas class="progressiveMedia-canvas js-progressiveMedia-
canvas" width="75" height="22"></canvas><img class="progressiveMedia-
image js-progressiveMedia-image" data-src="https://cdn-images-1.
medium.com/fit/t/800/240/1*zGHaOHYGV4LRCqSlyAtMBg.jpeg" style="
opacity: 1; visibility: visible"><noscript class="js-progressiveMedia
-inner"><img class="progressiveMedia-noscript js-progressiveMedia-
inner" src="https://cdn-images-1.medium.com/fit/t/800/240/1*
zGHaOHYGV4LRCqSlyAtMBg.jpeg"/></noscript></div></div></figure><h3
class="graf graf--h3 graf-after--figure graf--title" id="21fb" name="
21fb">5 Tricks to Master Pandas append()</h3><h4 class="graf
graf--h4 graf-after--h3 graf--trailing graf--subtitle" id="7163" name
="7163">Hands-on tutorial to effectively add rows at the end of
current...</h4></div></div></section></div></a></div><div class="
postArticle-readMore"><a class="button button--smaller button--
chromeless u-baseColor--buttonNormal" data-action="open-post" data-
action-source="---------2----------------------" data-action-value="
https://towardsdatascience.com/5-tricks-to-master-pandas-append-
ede4318cc700?source=---------2----------------------" data-post-id="
ede4318cc700" href="https://towardsdatascience.com/5-tricks-to-master
-pandas-append-ede4318cc700?source=---------2----------------------"
>Read more...</a></div><div class="u-clearfix u-paddingTop10"><div
class="u-floatLeft"><div class="multirecommend js-
actionMultirecommend u-flexCenter" data-is-flush-left="true" data-
post-id="ede4318cc700" data-source="listing-----ede4318cc700
---------------------clap_preview"><div class="u-relative u-
foreground"><button aria-label="Clap" class="button button--primary
button--chromeless u-accentColor--buttonNormal button--withIcon
button--withSvgIcon clapButton js-actionMultirecommendButton
clapButton--darker" data-action="sign-up-prompt" data-action-source="
listing-----ede4318cc700---------------------clap_preview" data-
redirect="https://medium.com/_/vote/p/ede4318cc700" data-requires-
token="true" data-sign-in-action="multivote"><span class="button-
defaultState"><span class="svgIcon svgIcon--clap svgIcon--25px is-
flushLeft"><svg class="svgIcon-use" height="25" width="25"><g fill-
rule="evenodd"><path d="M11.739 0l1.761 2.966L13.261 0z"></path><path
```

```
d="M14.815 3.776l1.84-2.551-1.43-.471z"></path><path d="M8.378 1.224
l1.84 2.551L9.81.753z"></path><path d="M20.382 21.622c-1.04
1.04-2.115 1.507-3.166 1.608.168-.14.332-.29.492-.45 2.885-2.886
3.456-5.982 1.69-9.211l-1.101-1.937-.955-2.02c
-.315-.676-.235-1.185.245-1.556a.836.836 0 01.66-.16c
.342.056.66.28.879.605l2.856 5.023c1.179 1.962 1.379 5.119-1.6 8.098m
-13.29-.528l-5.02-5.02a1 1 0 01.707-1.701c.255 0 .512.098.707.292l2
.607 2.607a.442.442 0 00.624-.624L4.11 14.04l-1.75-1.75a.998.998 0
111.41-1.413l4.154 4.156a.44.44 0 00.624 0 .44.44 0 000-.624l
-4.152-4.153-1.172-1.171a.998.998 0 010-1.41 1.018 1.018 0 011.41 0l1
.172 1.17 4.153 4.152a.437.437 0 00.624 0 .442.442 0 000-.624L6.43
8.222a.988.988 0 01-.291-.705.99.99 0 01.29-.706 1 1 0 011.412 0l6
.992 6.993a.443.443 0 00.71-.501l-1.35-2.856c
-.315-.676-.235-1.185.246-1.557a.85.85 0 01.66-.16c
.342.056.659.28.879.606L18.628 14c1.573 2.876 1.067 5.545-1.544
8.156-1.396 1.397-3.144 1.966-5.063
1.652-1.713-.286-3.463-1.248-4.928-2.714zM10.99 5.976l2.562 2.562c
-.497.607-.563 1.414-.155 2.284l1.265.562-4.257-4.257a.98.98 0
01-.117-.445c0-.267.104-.517.292-.706a1.023 1.023 0 011.41 0zm8.887
2.06c-.375-.557-.902-.916-1.486-1.011a1.738 1.738 0 00-1.342.332c
-.376.29-.61.656-.712 1.065a2.1 2.1 0 00-1.095-.562 1.776 1.776 0
00-.992.128l-2.636-2.636a1.883 1.883 0 00-2.658 0 1.862 1.862 0
00-.478.847 1.886 1.886 0 00-2.671-.012 1.867 1.867 0 00-.503.909c
-.754-.754-1.992-.754-2.703-.044a1.881 1.881 0 000 2.658c
-.288.12-.605.288-.864.547a1.884 1.884 0 000 2.659l.624.622a1.879
1.879 0 00-.91 3.161l5.019 5.02c1.595 1.594 3.515 2.645 5.408 2.959a7
.16 7.16 0 001.173.098c1.026 0 1.997-.24
2.892-.7.279.04.555.065.828.065 1.53 0 2.969-.628 4.236-1.894
3.338-3.338 3.083-6.928 1.738-9.166l-2.868-5.043z"></path></g></svg><
/span></span><span class="button-activeState"><span class="svgIcon
svgIcon--clapFilled svgIcon--25px is-flushLeft"><svg class="svgIcon-
use" height="25" width="25"><g fill-rule="evenodd"><path d="M11.738 0
l1.762 2.966L13.262 0z"></path><path d="M16.634 1.224l-1.432-.47-.408
3.022z"></path><path d="M9.79.754l-1.431.47 1.84 2.552z"></path><path
 d="M22.472 13.307l-3.023-5.32c-.287-.426-.689-.705-1.123-.776a1.16
1.16 0 00-.911.221c-.297.231-.474.515-.535.84.017.022.036.04.053.063
l2.843 5.001c1.95 3.564 1.328 6.973-1.843 10.144a8.46 8.46 0
01-.549.501c1.205-.156 2.328-.737 3.351-1.76 3.268-3.268 3.041-6.749
1.737-8.914"></path><path d="M12.58 9.887c
-.156-.83.096-1.569.692-2.142L10.78 5.252c-.5-.504-1.378-.504-1.879
0-.178.18-.273.4-.329.63l4.008 4.005z"></path><path d="M15.812 9.04c
-.218-.323-.539-.55-.88-.606a.814.814 0 00-.644.153c
-.176.137-.713.553-.24 1.566l1.43 3.025a.539.539 0 11-.868.612L7.2
6.378a.986.986 0 10-1.395 1.395l4.401 4.403a.538.538 0 11-.762.762L5
.046 8.54 3.802 7.295a.99.99 0 00-1.396 0 .981.981 0 000 1.394L3.647
```

```
   9.93l4.402 4.403a.537.537 0 010 .761.535.535 0 01-.762 0L2.89 10.696a
   .992.992 0 00-1.399-.003.983.983 0 000 1.395l1.855 1.854 2.763 2.765a
   .538.538 0 01-.76.761l-2.765-2.764a.982.982 0 00-1.395 0 .989.989 0
   000 1.395l5.32 5.32c3.371 3.372 6.64 4.977 10.49 1.126C19.74 19.8
   20.271 17 18.62 13.982L15.812 9.04z"></path></g></svg></span></span><
   /button></div><span class="u-relative u-background js-
   actionMultirecommendCount u-marginLeft5"><button class="button button
   --chromeless u-baseColor--buttonNormal js-multirecommendCountButton u
   -disablePointerEvents" data-action="show-recommends" data-action-
   value="ede4318cc700">12</button></span></div></div><div class="
   buttonSet u-floatRight"><button class="button button--dark button--
   chromeless is-touchIconFadeInPulse u-baseColor--buttonDark button--
   withIcon button--withSvgIcon button--bookmark js-bookmarkButton" data
   -action="add-to-bookmarks" data-action-value="ede4318cc700"><span
   class="js-remove-from-bookmarks u-hide"><span class="svgIcon svgIcon
   --bookmarkFilled svgIcon--25px is-flushRight"><svg class="svgIcon-use
   " height="25" width="25"><path d="M19 6c0-1.1-.9-2-2-2H8c-1.1 0-2
   .9-2 2v14.66h.012c.01.103.045.204.12.285a.5.5 0 00.706.03L12.5 16.85
   l5.662 4.126c.205.183.52.17.708-.03a.5.5 0 00.118-.285H19V6z"></path>
   </svg></span></span><span class="js-add-to-bookmarks"><span class="
   svgIcon svgIcon--bookmarkAdd svgIcon--25px is-flushRight"><svg class=
   "svgIcon-use" fill="none" height="25" width="25"><path d="M18 2.5a
   .5.5 0 011 0V5h2.5a.5.5 0 010 1H19v2.5a.5.5 0 11-1 0V6h-2.5a.5.5 0
   010-1H18V2.5z" fill="#292929"></path><path d="M7 7a1 1 0 011-1h3.5a
   .5.5 0 000-1H8a2 2 0 00-2 2v14a.5.5 0 00.805.396L12.5 17l5.695 4.396A
   .5.5 0 0019 21v-8.5a.5.5 0 00-1 0v7.485l-5.195-4.012a.5.5 0 00-.61 0
   L7 19.985V7z" fill="#292929"></path></svg></span></span></button></
   div></div></div>
               </div>
           </div>

</body>
```

Appendix D

Future Work

This section will describe future work that must be done in order to improve upon the current application.

D.1 Modeling

D.1.1 Fixing False Positives and False Negatives

As described in Section 5.5, there are a number of False Positives and False Negatives that do not seem reasonable for the model to make. It would be helpful to delve into this problem further and see if there is some further text processing that can be used to remove these errors.

D.2 Client

D.2.1 Streamline New User Integration

Currently, when a new user signs up, there is no guidance or immediate data fetching to show the user when they first sign-in to the application. This, of course, is not acceptable for a finished product. In the future, there would be a series of sequential pages that would be shown to the user to:

1. Ask for keyword(s)
2. Ask what sources they would like to see
3. Fetch a small amount of data immediately to show the user

This would create a much better experience for a new user.

D.3 Analytics

D.3.1 Model Performance Monitoring

Currently, there is no monitoring of user model metrics. It would be beneficial to track the performance of each user model, in aggregate, and determine how metrics perform over time with an increase in the number of user labels.

D.3.2 Data Download Monitoring

As mentioned in Appendix B, user data is currently downloaded using a loop over each user and their respective sources. As the number of users, and potentially sources, grows, this can cause bottlenecks in the amount of time it takes to download new data every day. For this reason, it would be helpful to monitor metrics such as the amount of time it takes to download data per user, per source, etc. By leveraging these metrics, we can determine when more servers would be needed, to download user data, as the user base grows.

D.4 Database

D.4.1 Table Denormalization

As mentioned in Section A.3.5, normalization was used in the database to reduce redundant information for the data that was downloaded from each source. Currently, a ‘data‘ table and a ‘user_data_relation‘ table was used to implement normalization; however, in the future it may be worthwhile to remove this normalization, at the expense of higher storage costs, to reduce the computational cost of joins when retrieving data for a given user.

Glossary

API Application Programming Interface is used as a standardized method of communication with software 100, 107

authentication resource The REST endpoints used for authentication purposes for the application 103

backend The component of the application that handles all communication between the client and the database and REST calls 10, 100, 103, 117, 134, 135

client The user facing component of the application 115, 135

data traversal resource The REST endpoints used for passing data between the client and backend of the application; typically through user inputs 103

followee Who a given twitter user is following 17, 136, 141, 144

GET A type of REST request used to receive messages from a server 107, 109, 114–116, 123

JSON A JavaScript Object Notation File 104, 126

POST A type of REST request used to send data to add to a server 104–106, 108, 114–118, 124

proxy dataset A dataset comprising of tweets and reddit posts with a label for positive or negative sentiment and arxiv papers with a label indicating if it is from the computer science category or not 22, 23, 31, 32, 48, 49, 58, 62, 68

PUT A type of REST request used to update data on a server 110, 112, 119, 120, 122

sample dataset The dataset that was obtained by Professor Kimon Fountoulakis manually labeling posts from Twitter, Reddit, Arxiv and PMLR. The total size of this dataset was 647 datapoints 22, 23, 30–32, 48, 54, 57, 58, 60, 93

sources A hash-map with a key-value pair representing a source and settings for that source. This is a column in the ‘users‘ table of the database. 125, 126

sources_metadata A hash-map with a key-value pair representing a source and metadata for that source. This is a column in the ‘users‘ table of the database 125, 126, 136