Quan Vo, Van Vo

Predicting posts’ popularity from r/Jokes and r/askreddit

Abstract:

This paper explores the use of Recurrent Neural Networks (RNNs) to classify Reddit posts as popular or unpopular. Two variations of the RNN model were investigated: one using only post text as input and another incorporating the post's creation time. Additionally, topic modeling was employed to identify the most popular topics on Reddit, providing valuable insights into the data and informing the model development process. For r/askreddit, the text-only model reached 69.76% on test accuracy, while text and time model reached 63% test accuracy. For r/jokes, text-only model reach 66.31% on test accuracy, while text and time model 66.45% test accuracy

1. Introduction:

The rise of social media platforms like Reddit has fundamentally changed the way we consume and share information. With millions of users generating and engaging with content daily, understanding the factors that contribute to a post's success becomes increasingly important. This research examines the potential of Recurrent Neural Networks (RNNs) to identify and classify Reddit posts as popular or unpopular. Our research questions are: 1/ Which topics are the most popular on r/askreddit and r/jokes?, 2/ Can RNNs effectively predict the popularity of a reddit post? And 3/ Does the creation time of a post affect its popularity?

RNNs have established themselves as a powerful tool for analyzing sequential data, such as text and time series. Their ability to learn and exploit long-term dependencies within sequences makes them ideal candidates for tasks like sentiment analysis, language translation, and, as investigated in this study, post popularity classification.

This study employs two variants of an RNN model: one utilizing only the textual content of a post and another incorporating both the text and the timestamp of its creation. This comparative approach allows for a deeper understanding of the relative importance of each input feature in determining a post's popularity. In particular, we want to find out whether published time affects the popularity of a post.

Furthermore, the research incorporates a topic modeling pre-processing step. By analyzing the underlying thematic structure of Reddit content, we can identify the most prevalent and engaging topics, providing valuable insights into user preferences and potential biases within the data.

The findings of this study contribute to our understanding of user behavior on social media platforms and offer valuable insights for content creators aiming to maximize engagement. Additionally, the insights derived from topic modeling can be leveraged to improve the effectiveness of recommendation systems and personalize user experiences on online platforms.

2. Related Work:

There was some previous attempt to predict the popularity of reddit posts and comments. Weller and Seppi (2019) train a Transformer to classify posts from r/jokes into humorous or unhumorous[1]. Their model reaches 72.4% in accuracy. Zayats and Ostendorf (2019) used graph-structured Bidirectional LSTM to predict popularity of comments from comment threads[2]. They achieved an F1 score of 56.4% for r/askwomen, 54.8% for r/askmen, and 50.4% for r/politics.

3. Data:

Reddit is a well-known social media platform consisting of a large number of subreddits of various topics and interests. In this research, we experimented on two popular subreddits: r/askreddit and r/Jokes. We obtained data of r/askreddit's and r/Jokes' posts from two main sources: Existing datasets and new data scraped by ourselves using PullPush.

3.1 r/askreddit:

3.1.1 source:

The r/askreddit dataset is created partly by data collected from an existing dataset: Social Grep dataset at Hugging Face: one-million-reddit-questions[5]. We also scrape more, newer data ourselves using PullPush[3] - a service for indexing and retrieving data from reddit.

3.1.2 analysis:

The Social grep dataset contains one million reddit posts collected from subreddit r/askreddit, in which each data point contains multiple fields related to a post such as title, score, creation time, etc. We also scrape more data from the subreddit itself using PullPush. After that, we combined all data together and annotated them as Popular or Unpopular to create the final dataset. Figure 1 shows the distribution of score in the r/askreddit dataset.

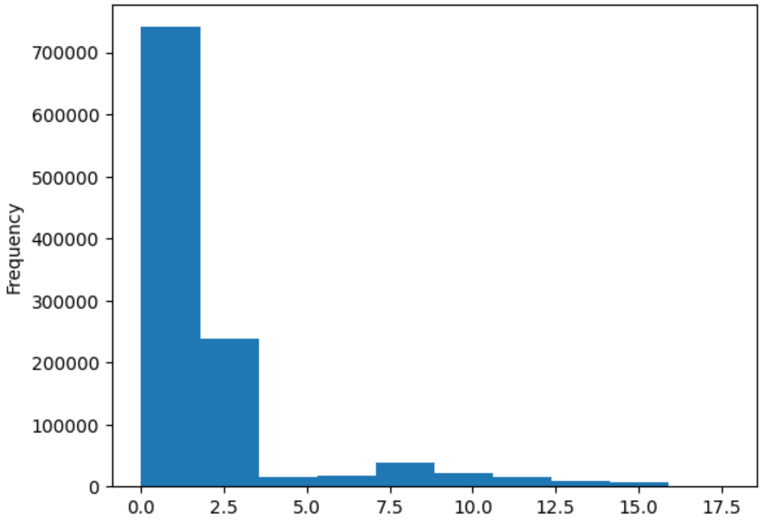

Figure 1: Distribution of scores (log2) of r/askreddit dataset

3.2 r/Jokes:

3.2.1 source:

The r/jokes dataset is created by combining data collected from publicly available dataset from paper: The rJokes Dataset: a Large Scale Humor Collection[4]. The number of highly scored data points is low, so we also scrape more data from the subreddit itself using PullPush.

3.2.2 analysis:

The rJokes Dataset consists of over 550,000 jokes posted over an 11 year period on r/Jokes subreddit. There are 4 important fields in the dataset: Punchline, Body, Joke and Score. Joke is the combination of both Punchline and Body. Score is the difference between number of upvotes and downvotes a post received. Figure 2 shows the distribution of log2 score in the r/Jokes dataset.

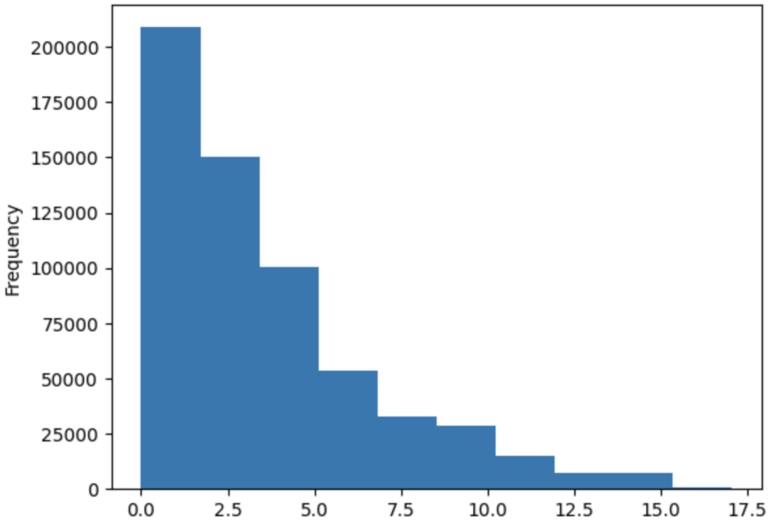

Figure 2: distribution of scores (log2) of r/jokes dataset

4. Methodology:

4.1 Models:

We employed a Bidirectional LSTM architecture as the foundation of our model, investigating the impact of post creation time through two distinct model variants. The first variant relied solely on the text content as input, while the second introduced the hour of creation as an additional feature. This comparative analysis aimed to elucidate the role of creation time in predicting post popularity.

4.2 Training:

Firstly, we balanced the data between popular and unpopular labels. Then, we split the dataset into 70% train data, 20% validation data and 10% test data. The models were trained for 10 epochs with batch size 32. We used Adam optimizer with an initial learning rate of 0.0001.

We use accuracy as our evaluation metrics.

4.3 Topic Modeling:

Explainability has always been an important challenge for machine learning models. In order to partly address this issue in our research, gain better insights into how a post's topic might influence its own popularity and acquire a deeper understanding of the two subreddits' data, we decided to implement topic modeling using BERTopic. BERTopic is a topic modeling technique that leverages various embedding techniques and c-TF-IDF to create dense clusters allowing for easily interpretable topics. In this research, we employed the BERTopic model on data points annotated as Popular in our datasets to find out which topics tend to be more popular and attract more interactions from the users. Table 1 shows our BERTopic parameters

Sentence Transformer	all-MiniLM-L6-v2
UMAP	n_neighbors=15, n_components=5, min_dist=0.0, metric='cosine'
HDBSCAN	min_cluster_size=300, metric='euclidean', cluster_selection_method='eom'

Table 1: BERTopic parameters

4.3.1 r/askreddit:

For the r/askreddit dataset, BERTopic model detected 58 different topics. Table 4 shows the top 10 most popular topics from r/askreddit.

Topic Name	Mean Score
35_illegal_legal_illegally_legally	6341.36
33_subreddits_subreddit_sub_subs	6187.03
15_protagonist_character_characters_fictional	5910.69
25_sleep_sleeping_insomniacs_asleep	5610.30
18_paranormal_haunting_haunted_ghosts	5393.73
8_teacher_teachers_classroom_students	5300.45
28_hate_dislike_hatred_hates	5296.68
-1_life_reddit_death_things	5222.64
1_movies_movie_films_film	5201.58
19_moment_situation_swore_worst	5132.51

Table 2: Top 10 most popular topics from r/askreddit

4.2.1 r/jokes:

For the r/jokes dataset, BERTopic model detected [] different topics. Table 5 shows the top 10 most popular topics from r/jokes.

Topic Name	Mean Score
56_batman_robin_bat_bats	8118.12
84_eclipse_solar_sun_moon	8002.92
41_vegans_vegan_vegetarian_meat	7379.48
5_removed_deleted_user_banned	6754.98
71_threesome_foursome_twosome_trois	6356.78
37_librarian_librarians_penises_library	6344.83
58_eggs_butter_cooking_cook	6245.81
59_korea_koreans_korean_seoul	6213.83
54_lumberjack_forest_dialogue_tree	5921.73
45_knights_knight_kingdom_king	5810.11

Table 3: Top 10 most popular topics from r/joke

5. Result:

5.1 r/askreddit:

Text-only model reached 76% accuracy on training data, 67% on validation and 69.76% on test data.

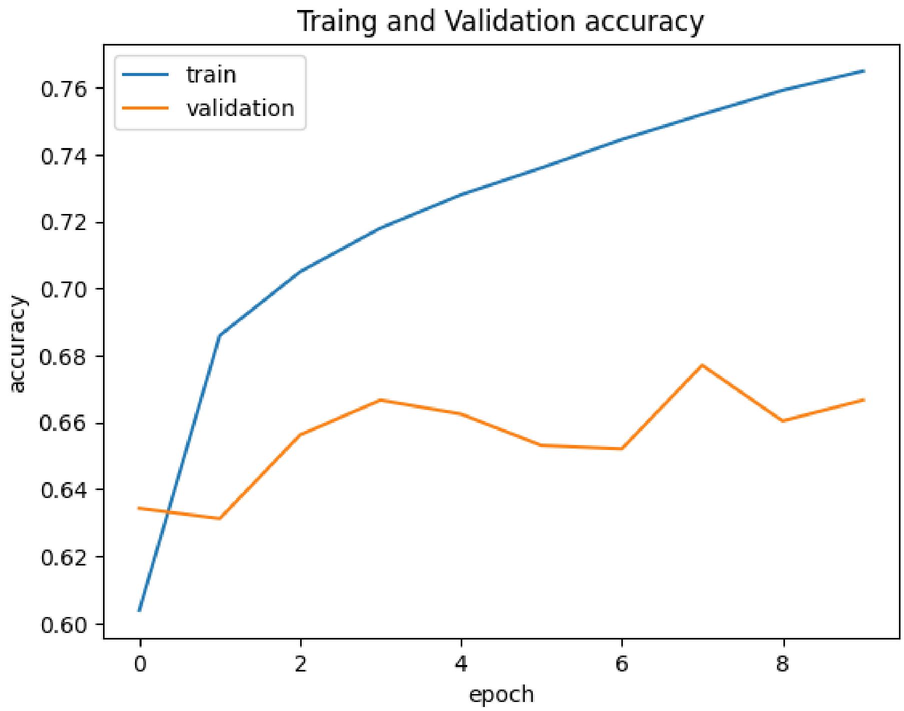

Figure 3: Training and Validation Accuracy Training History For Text-only Model

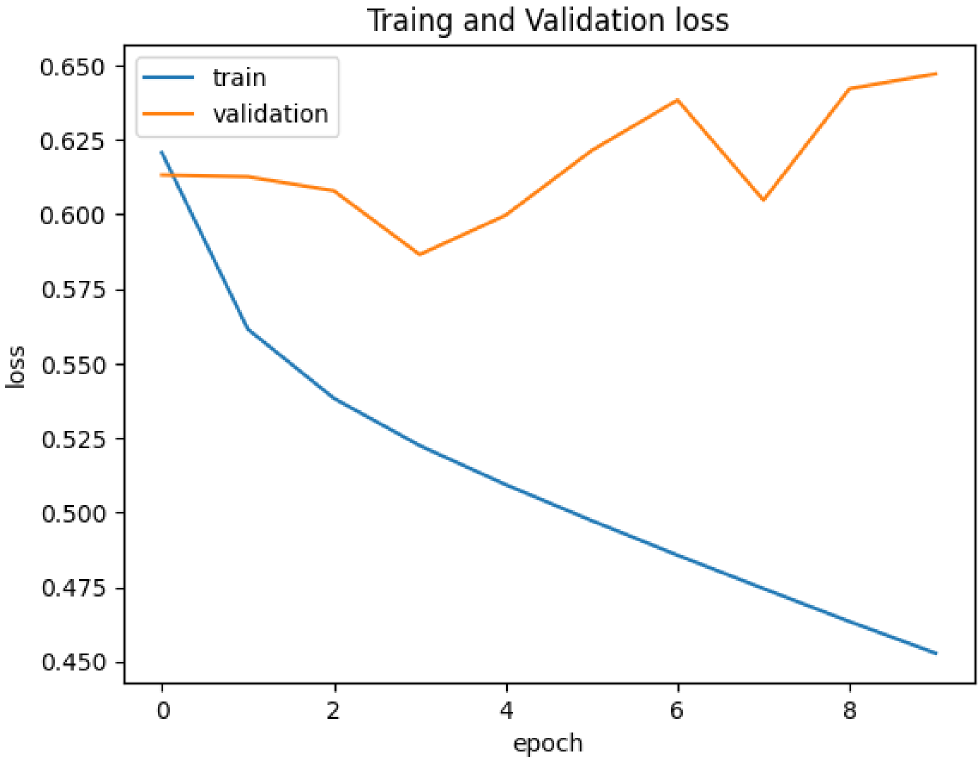

Figure 4: Train and Validation Loss History of Text-only Model

Text and created hour model reached 77% accuracy on training data, 61.5% on validation and 63% on test data.

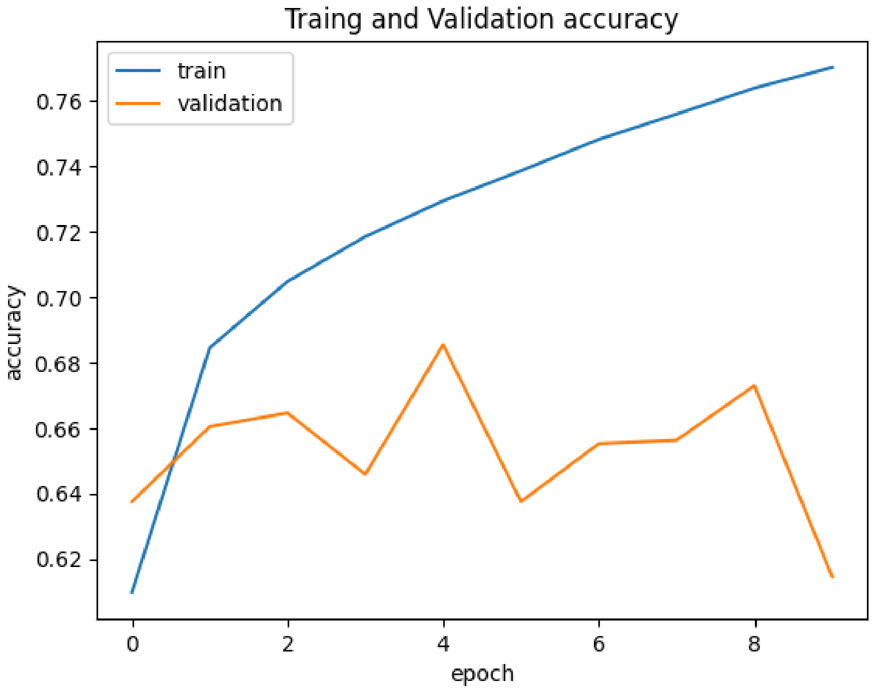

Figure 5: Training and Validation accuracy

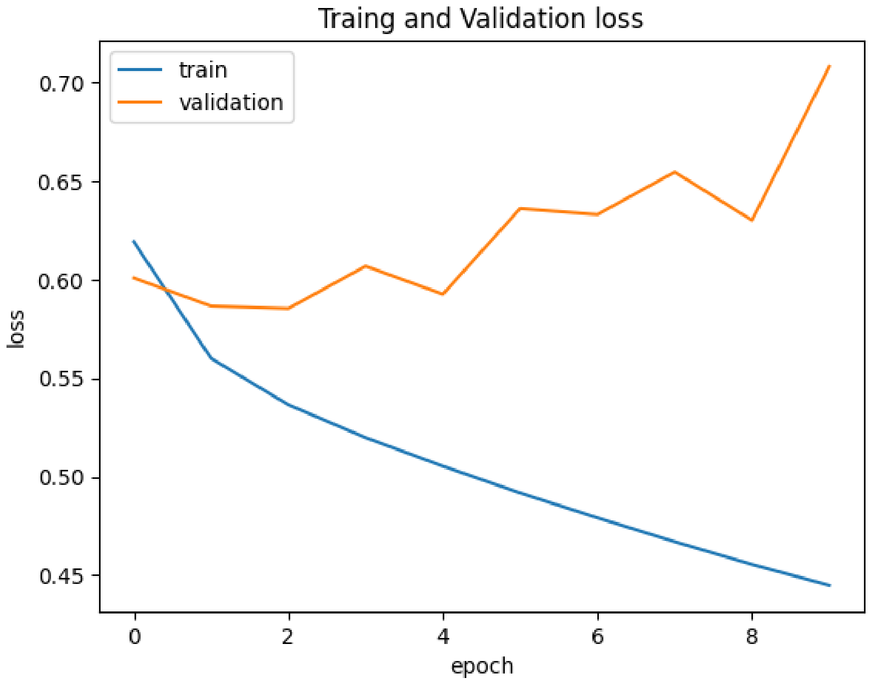

Figure 6: Training and Validation accuracy

5.2 r/jokes:

Text-only model reached 85.4% accuracy on training data, 71.04% on validation and 66.31% on test data.

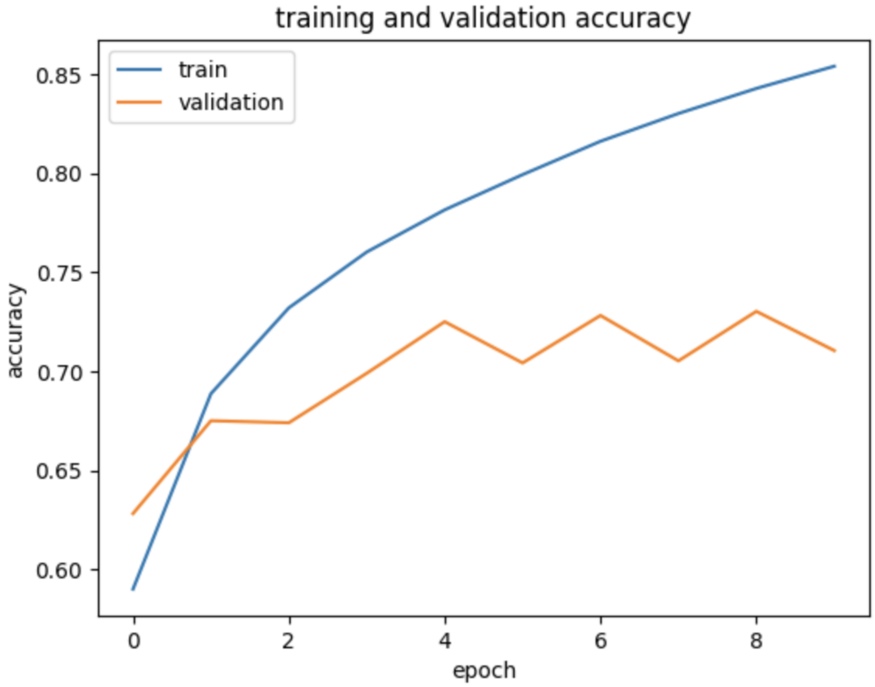

Figure 7: Training and Validation accuracy of text only model

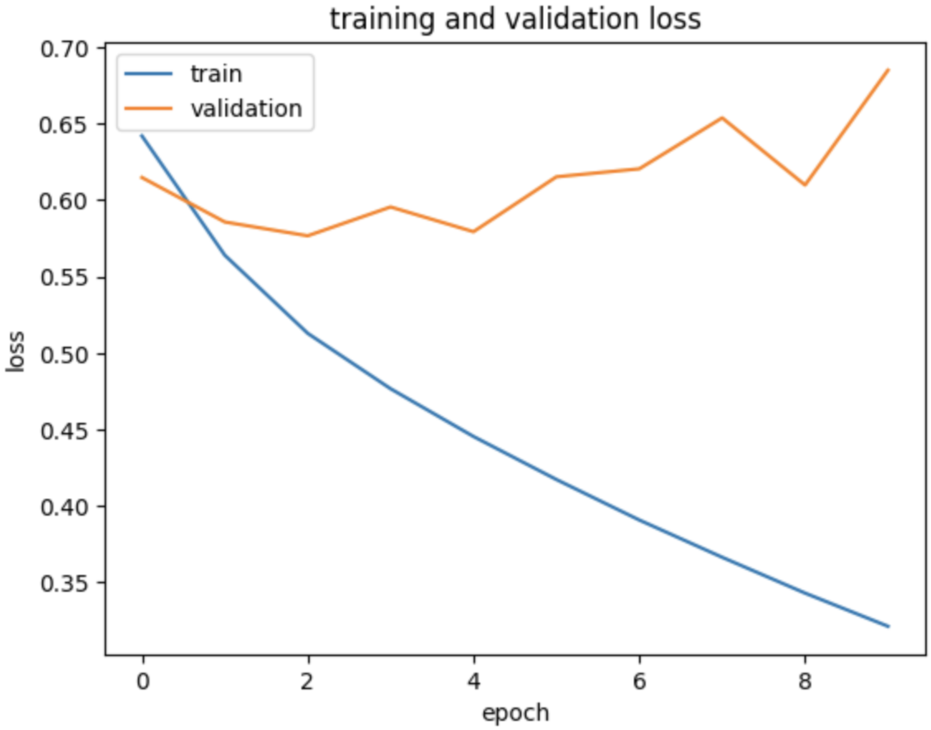

Figure 8: Training and Validation loss of text only model

Text and created hour model reached 84.61% accuracy on training data, 72.19% on validation and 66.45% on test data.

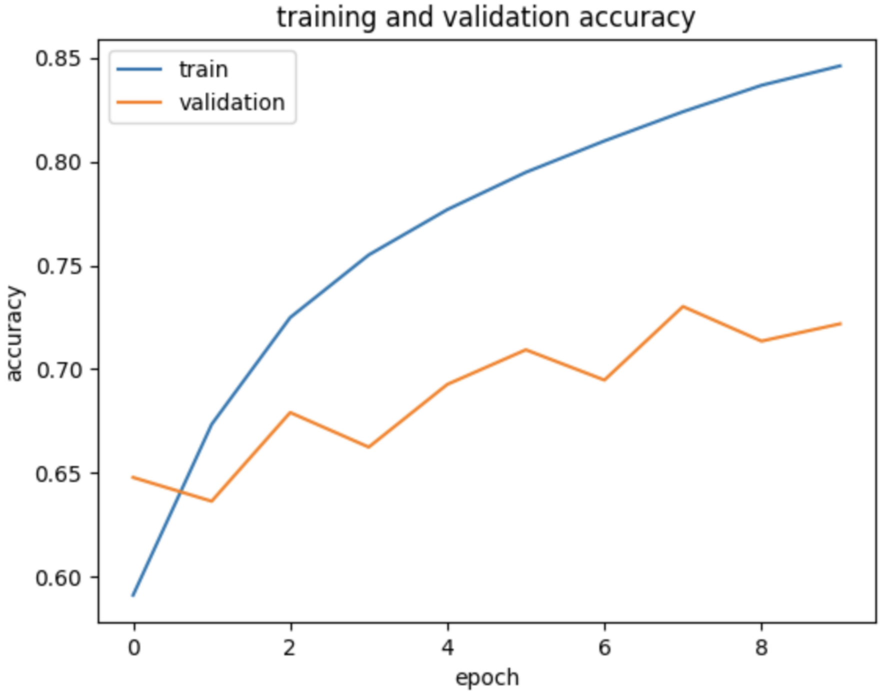

Figure 9: Training and Validation accuracy of text and created time model

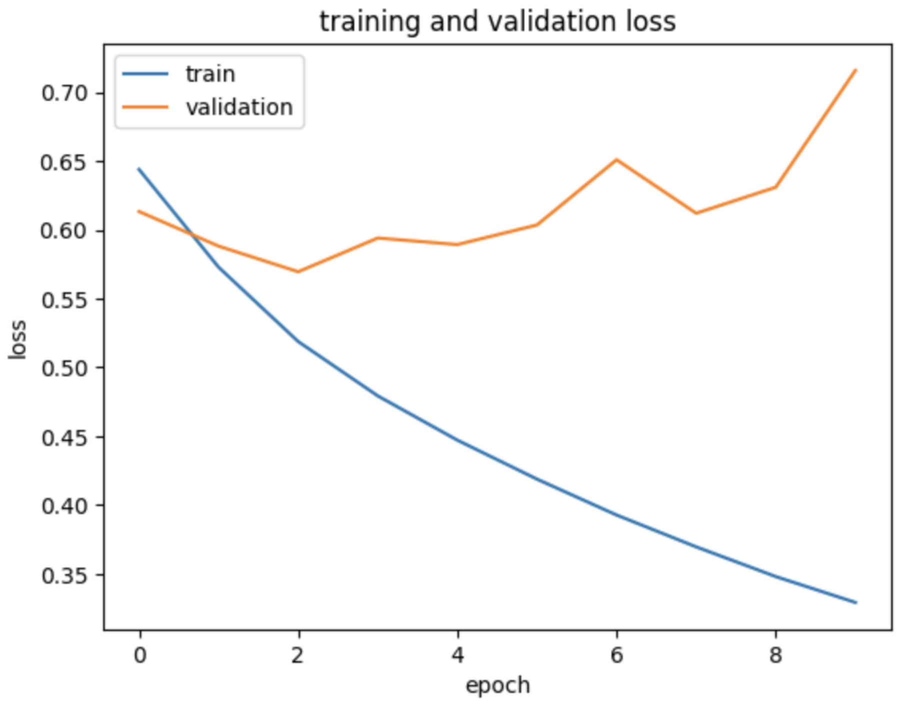

Figure 10: Training and Validation loss of text and created time model

6. Discussion and Future Work:

In this paper, we investigated how effective RNNs model predict popularity of reddit posts based on content and published time. Based on the performance of our models, it seems like we need more information than content and created time of a post to effectively predict the popularity of a post.

In our opinion, the popularity of a reddit post does not solely depend on its contents, but also on the first few interactions of the post. There is a chance that people will more likely interact with a post if it previously received positive interaction. For future work, we could expand our scope to include more information, such as some of the first comments or voting history when the posts are created. Another way we could expand our work is to rework on our definition of popular post. Currently, We decided that a post with a high score is popular. However, a post with a low score but has a high amount of total interactions such as the total number of comments plus upvotes can arguably be called popular.

References

[1] Weller, O., & Seppi, K. (2019). Humor detection: A transformer gets the last laugh. *arXiv preprint arXiv:1909.00252*.

[2] Zayats, V., & Ostendorf, M. (2018). Conversation modeling on Reddit using a graph-structured LSTM. *Transactions of the Association for Computational Linguistics*, *6*, 121-132.

[3] *Pullpush Reddit API documentation*. PullPush Reddit API. (n.d.). https://pullpush.io/

[4] Weller, O., & Seppi, K. (2020, May). The rjokes dataset: a large scale humor collection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference* (pp. 6136-6141).

[5] SocialGrep. *SocialGrep/one-million-reddit-questions · datasets at hugging face*. SocialGrep/one-million-reddit-questions · Datasets at Hugging Face. https://huggingface.co/datasets/SocialGrep/one-million-reddit-questions