#PleaseUpVote: Virality Prediction with Title and Thumbnail Image on Reddit

Xiaowen Lin
Stanford University
veralin@stanford.edu

Zijian Wang
Stanford University
zijwang@stanford.edu

Chaonan Ye
Stanford University
yec0214@stanford.edu

Abstract

Reddit is the 5th most visited website in the United States[1]. A campaign or advertisement post can potentially reach millions of target audience. However, there is little work on predicting the virality, or popularity, of Reddit posts. This is indeed a hard task because even humans could hardly distinguish between two posts with different popularity. Here, we propose to train a multimodal neural network based on titles and thumbnails. Our results show that our model is able to capture the nuance signal of virality, and joining image and text information yields the best result.

1 Introduction

Reddit, as one of the most visited discussion forums in the United States, is highly influential. Given the popularity of the site, the exposure of the top posts could be extremely high, which means a campaign or advertisement post can potentially reach millions of target audience. An effective virality prediction model would be crucial in curating a popular post. Further, it will benefit many downstream studies and tasks including marketing, data mining and automated monitoring of the spread of viral rumors.

As illustrated in Figure 1[2], the title and the thumbnail are two of the most important factors of the virality of a post. Thus, we propose to train a multimodal neural network to predict the virality of the post based on these two features. This is a challenging task because even for humans, it is difficult to accurately judge if a post will go viral or not. Further, there is little previous work and no fine-grained dataset. In this work, we i) identified subreddits that require an image in the submission, and created the dataset, ii) train image, text, and multimodal models, and iii) show that our models could capture the nuance signal of virality, and joining image and text information yields the best result.

Figure 1: An example submission from the main page view

[1] https://www.alexa.com/siteinfo/reddit.com
[2] A full example is shown in Appendix.

2 Related Work

Predicting virality is a hard problem. Topics, in general, often have a periodic effect as they peak and decline in virality. Further, it is affected by various extraneous factors other than the topic itself, for example, the community norm or posting time [15]. Such prediction on the reddit has been known as an even harder task. Within a crowdsourced annotation analysis of over 7600 posts, researchers found that given two posts with thumbnails and titles with similar scores, people could distinguish which one has a higher score only with an accuracy of 0.525, barely better than a random choice [3].

Most previous works on virality prediction employed traditional machine learning and/or network-based algorithms [4, 18]. Later, some literature studied such problems using deep learning techniques, but most of them only focused on a single source, i.e., either image or text [2, 6, 5, 1]. Most recently, researchers found that taking both information into account using multimodal models helps improve performance on social networks like Instagram [11] and Twitter [17]. However, few of them has focused on anonymous social networks such as Reddit. Out of those who researched on Reddit, most of them were using traditional machine learning techniques, e.g., Term Frequency–Inverse Document Frequency (TF-IDF) [15], Latent Dirichlet Allocation (LDA) [15], Support Vector Machine (SVM) [16] or Naïve Bayes [16]. However, those techniques fail to take advantage of either the vast amount of data available online or the potential co-effect of both image and text data on virality prediction. In order to exploit various data sources and improve the poor performance of virality prediction on Reddit, we plan to investigate the efficiency of using multimodal deep learning techniques to predict the virality of posts on Reddit.

3 Data

3.1 Data Collection

Among popular subreddits, four of them (`aww`, `EarthPorn`, `politics`, and `The_Donald`) always have mandatory thumbnails within posts. Thus, we extracted submissions in these subreddits from 2015 to 2018. Titles are directly available from the json dump of submissions, and images were crawled on our own. To filter out noises, we excluded the submissions created within 7 days of the extraction time. Moreover, we filtered out whose images were actually system-generated screenshots from videos. The statistics of the datasets is shown in Table 1.

Subreddit	Topic	Number of Entries	Number of Avail. Images
`aww`	Cute pictures of animals	1,720,414	1,076,371
`politics`	Politics	1,432,923	545,644
`The_Donald`	Donald Trump	4,573,934	1,858,266
`EarthPorn`	Landscape photography	280,745	129,251

Table 1: The topics and statistics of the datasets

3.2 Data Prepocessing

For image, as thumbnails are usually in small size (e.g., 140 x 140), we scaled everything up to 224 x 224 with the white padding. For text, we tokenized the title using SciPy [10]. Moreover, we transform the raw score to logarithmic scale. Finally, we partitioned data into 90% training, 5% development, and 5% test with stratification. The training dataset was balanced via oversampling.

4 Task and Methods

4.1 Task

The aim of this project is to predict the virality of a Reddit submission after a certain range of time. Such virality may be directly inferred from the title and the thumbnail of a submission, as they are

what people see at the very first time. Hence, we want to investigate what effects the title and thumbnail have on the virality of a submission.

To do this, we trained a multimodal deep learning model that takes advantage of both image and text information. Before joining images and texts, two separate models, referred as baseline models, were trained as a pre-training step for the final model. More details will be described later in this section.

4.2 Methods

We present the methods used for the construction of the image, text, and multimodel models. A high level architecture sketch is shown in Figure 2.

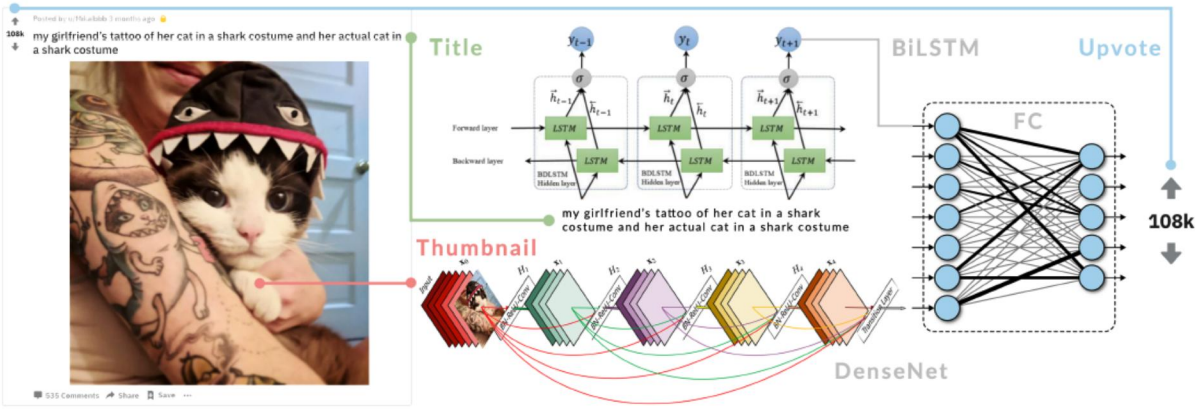

Figure 2: The network architecture of multimodel model

Image We choose some of the state-of-the-art models. Specifically, we set our baseline as ResNet-18 [6]. We further investigated different network model architectures and layer sizes, and found that a more complicated and deeper model, DenseNet-161[8], yields better performance. A full list of hyperparameters we tried will be presented later in the next section.

Text The text model was built using an embedding layer, following by a 2-stack bi-directional Long short-term memory (LSTM) layer [7]. Batch normalization [9] was applied afterwards with a couple of fully connected layers to achieve better convergence. Dropout was also applied between the two LSTMs to prevent overfitting. We tried to use different pretraining strategies for embeddings, which would be listed in the next section.

Multimodal After finalizing the two models mentioned above, we trained a multimodal model that concatenated the fully connected layers from image and text models with a few fully-connected layers on top.

5 Experiments, Results, and Discussions

5.1 Classification

As an initial step, we formulated the task as a two-class classification problem, where we classified the posts less than `mean - std` upvotes as negative, and more than `mean + std` as positive based on the distribution of a specific subreddit. We followed the models described in §4. The models were evaluated based on Macro-F1 on a held-out test dataset. Note that in this section we focus on the subreddit `aww` as i) it has a decent amount of data, and ii) it is less temporal.

5.1.1 Hyperparameter Tuning

For this task, a list of hyperparameters needs to be tuned to achieve the best performance. We list the major ones[3] used for all models in Table 2[4].

Hyperparameter	Options
Data	**Oversampled** / Raw data
Image Model	DenseNet (121, **161**) / ResNet (18, 101, 152) ; **Pretrained** / None
Text Model	Hidden size (150, **300**); Dropout ratio (0.2, **0.5**); **Bi-Dir.** / Single-Dir.
Text Embedding	Pretrained: Word2Vec [12] / GloVe [14]; Non-pretrained: Character-level / **Word-level**
Optimizer	**SGD with momentum** / Adam / AMSGrad
Batch size	Image & text: **64**, 128; Multimodal: 16, **32**

Table 2: Overview of the major hyperparameters used for models. Bold options are the ones that perform the best.

5.1.2 Results and Discussions

In Table 3, we list selected results for the classification models.

	Method	Macro-F1
Baseline	Random	0.494
	Majority	0.360
Image	ResNet-18 + P + SGD	0.678
	DenseNet-161 + NP + SGD	0.653
	DenseNet-161 + P + SGD	**0.702**
	DenseNet-161 + P + Adam	0.681
Text	LSTM + NP + SGD	**0.618**
	LSTM + W2V + SGD	0.603
	LSTM + Glove + SGD	0.617
	LSTM + Char. + SGD	0.609
	Multimodal	**0.738**

Table 3: Performance comparisons between different models over held-out test set for aww. In the table, P denotes pretrained and NP denotes non-pretrained.

We demonstrate that there are sufficient cues to distinguish low and high-virality posts on some subreddits. Further, the multimodal model performs the best (lower section), and images give stronger signals (upper section) than textual data (middle section). On the image side, we show that small vision models (row 1) could capture virality signals well, while complicated ones do better (row 2-4). On the text side, using pretrained word embeddings does not lead to the better performance. The reason could be that the vocabulary on Reddit is different from the general text (Google News for Word2Vec[5] and Wikipedia for Glove[6].

5.2 Regression

The classification task does not take all potential information into consideration (e.g., posts with 1,000 and 10,000 scores were treated as the same). Intuitively, a regression model should be the next step to test out. Arguably, the regression is harder than the classification. To this end, we crawled the EarthPorn subreddit. Due to its focus on landscapes and its relatively small data amount, we are able to crawl high resolution images.

[3] Due to limited space, full details of each hyperparameter could be seen in the code release.

[4] All pretrained models, optimizers, and loss functions used are from the implementation in Pytorch 1.0 [13].

[5] http://mccormickml.com/2016/04/12/googles-pretrained-word2vec-model-in-python/

[6] https://nlp.stanford.edu/projects/glove/

We use the same architectures mentioned in above sections, except for the final output layer, which is a sigmoid function predicting a normalized upvote scores ranging from 0 to 1. The models were trained using mean squared error (MSE) loss. The initial results are shown in Table

Method	**MSE**
Random	0.158
Majority	0.091
Image	**0.035**
Text	0.154
Multimodal	0.038

Table 4: Performance of regression models for `EarthPorn`.

The results show that images yield a way stronger signal of virality, suggesting that i) this subreddit is more related to images than to text ii) high resolution images probably play an important role in such subreddits. However, the results compared to the baselines show that text might not be related to virality, as it is only slightly better than a random regressor. Further, its irrelevance slightly affects performance of the multimodal model. This might be due to class imbalance and limited number of posts, which makes it insufficient to build a good vocabulary. As limited by time and computational resources, we did not perform a thorough investigation on potential improvements. A few next steps will be included in §6.

6 Future Work

We outline a few future work that may be helpful to improve the performance of virality prediction on Reddit.

- Collect datasets with high resolution images: thumbnails are small, but people may click into the post and view the original image. Though beneficial, this requires more computational resources and needs more time for image crawling.
- Model temporal features: virality is highly related to temporal features. For example, the same post posted at weekend evening or workday morning may have significantly different virality. To better model such information, it might be possible to add temporal features (e.g., month, weekday/weekend, post time) into the model. This may also benefit those time-sensitive subreddits, e.g., `politics` and `The_Donalds`, where our current model is barely better than random choice.
- Regressions: in this work, we present simple regression results. However, there are still plenty of room for improving performances, e.g., a specifically-defined loss, evaluation metrics, and better class balancing strategies.
- Human performance annotation and comparison: in order to evaluate the models, a comparison to the human performance is needed. However, as different people have different views in terms of "virality", the annotation requires a diverse set of people. Further, the annotation questions need careful design to avoid noises from the annotators (e.g., they may randomly choose virality/non-virality if it is a two-choice question).

7 Conclusion

Analyzing the virality of a Reddit submission is a difficult perception task. It is hard even for humans to distinguish such a subtle attribute. Here, we investigate deep learning methods to model the virality. We crawled and built our own dataset for four subreddits. Then, we demonstrate that there is a correlation between a submission's thumbnail and title with its virality. Our results suggest that our model is able to capture the nuance signal of virality, and joining image and text information yields the best result. Finally, we propose a few directions for future improvements of the task. The code and dataset produced in this work will be released at `https://github.com/zijwang/cs230-project`.

Contributions

X.L. worked on data collection, poster and paper writing, regression models, and team discussions. Z.W. worked on data collection, poster and paper writing, text model construction, team discussions, and miscellaneous arrangement. C.Y. worked on data collection, poster and paper writing, image model construction, and team discussions. We believe the work was divided equally.

Acknowledgements

We thank Pedro Garzon, Ahmadreza Momeni, and all CS230 staff for their suggestions and support through this project.

References

[1] Xavier Alameda-Pineda, Andrea Pilzer, Dan Xu, Nicu Sebe, and Elisa Ricci. Viraliency: Pooling local virality. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[2] Arturo Deza and Devi Parikh. Understanding image virality. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1818–1826, 2015.

[3] Greg Stoddard. Understanding Popularity on Reddit. `https://medium.com/@gregstod/guess-the-karma-2-0-82a224a691f3`, 2016. Online; accessed 11 November 2018.

[4] Marco Guerini, Carlo Strapparava, and Gözde Özbal. Exploring text virality in social networks. In *ICWSM*, 2011.

[5] Ji He, Mari Ostendorf, and Xiaodong He. Reinforcement learning with external knowledge and two-stage q-functions for predicting popular reddit threads. *arXiv preprint arXiv:1704.06217*, 2017.

[6] Ji He, Mari Ostendorf, Xiaodong He, Jianshu Chen, Jianfeng Gao, Lihong Li, and Li Deng. Deep reinforcement learning with a combinatorial action space for predicting popular reddit threads. *arXiv preprint arXiv:1606.03667*, 2016.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[8] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, volume 1, page 3, 2017.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[10] Eric Jones, Travis Oliphant, and Pearu Peterson. {SciPy}: open source scientific tools for {Python}. 2014.

[11] Masoud Mazloom, Robert Rietveld, Stevan Rudinac, Marcel Worring, and Willemijn Van Dolen. Multimodal popularity prediction of brand-related social media posts. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 197–201. ACM, 2016.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[13] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[14] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[15] Tracy Rohlin. Popularity prediction of reddit texts. 2016.

[16] Jordan Segall and Alex Zamoshchin. Predicting reddit post popularity. 2016.

[17] Ke Wang, Mohit Bansal, and Jan-Michael Frahm. Retweet wars: Tweet popularity prediction via dynamic multimodal regression. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1842–1851. IEEE, 2018.

[18] Lilian Weng, Filippo Menczer, and Yong-Yeol Ahn. Predicting successful memes using network and community structure. In *ICWSM*, 2014.