Virality of Content on Social Media

Student Name: Aman Agarwal
Roll Number: 2015012

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Engineering

on November 22, 2018

BTP Track: Research

BTP Advisor
Dr. Ponnurangam Kumaraguru

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **Virality of Content on Social Media** submitted by me for the partial fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Engineering* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr. Ponnurangam Kumaraguru**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

.... **Place & Date:** *New Delhi, November 22, 2018*
Aman Agarwal

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

.. **Place & Date: ....**
Dr. Ponnurangam Kumaraguru

Abstract

Virality of online content on social networking websites is an important but esoteric phenomenon often studied in fields like marketing, psychology and data mining. In this project, I have studied viral content from a computer vision and natural language perspective. I have introduced a new dataset from Twitter for studying virality and defined an annotation score using Twitter metadata. I have also trained a deep learning *Siamese* model to predict virality of individual posts using relative virality in pairs of posts.

Keywords: deep learning for the web, convolutional neural networks, image virality, image attributes, text virality

Acknowledgments

I would like to thank Dr. Ponnurangam Kumaraguru and his Research Assistants Gurpreet Singh Bhatia and Shubham Gupta for their constant support in doing this project. Their guidance has enabled me to find my interests in the field of studying and analyzing various aspects of social systems. If it wasn't for their guidance and the involving discussions that we engage in, the progress in the project would have been difficult. I thank them for showing me the way to do good research.

Work Distribution

All the work mentioned in this report has been done in the Winter and Monsoon Semesters of 2018. Broadly, the work related to vision was done in the winter semester while the work related to text was done in the monsoon semester.

Contents

Chapter 1

Introduction

How do things become viral on the Internet? Since people want their messages to spread in the most effective and efficient way possible, this question has received a great deal of attention, particularly in recent years, as we have seen a dramatic growth of social networking on the Web. Generally speaking, virality refers to the tendency of a content either to spread quickly within a community or to receive a great deal of attention by it. In studying the spreading process, I will focus on the content and its characteristics rather than on the structure of the network through which the information is moving.

Popular content on social media reflects the mindset of the people. Understanding and predicting the virality of this content can be of interest to law enforcement agencies for functioning more efficiently, among other applications. This calls for a data driven approach that filters "recreational" content and focuses on the virality of more important information on Social Media. The aim of this project is to come up with a methodology to predict the virality of posts based on their content (both caption and attached picture).

The report is structured as follows: first, we review previous works addressing the topic of virality in social networks, and particularly some focusing on content impact. Then, after describing the dataset collected and used for this work, we will proceed with the study of virality of content using a deep learning architecture.

Chapter 2

Background and Related Work

A variety of different disciplines from computational social science to computer vision have approached the problem of predicting content popularity on the Internet. Until recent years, most existing works [1, 2, 3] studied how people share content on social networking sites after it has been posted. They used the network dynamics soon after the content has been posted to detect an oncoming snowballing effect and predict whether the content will go viral or not. It can be argued that predicting virality after the content has already been posted is too late in some applications. In this project, we are interested in understanding the relations between the content itself (even before it is posted online) and its potential to be viral.

In the past decade, there have been studies that attempt to predict virality based on the content of the post. Although, most of these works analyzed 'text' [4, 5] and 'videos' [6, 7, 8] only. There is a growing body of work on predicting image popularity and virality [9, 10], with the work of Khosla et al [10], which focuses on introducing the problem of predicting image popularity, and then presenting a straightforward formulation with image features applied to a classifier. Deza and Parikh [9] introduce the different problem of predicting image virality with a novel metric for identifying virality, and demonstrate the effectiveness of several techniques for solving the problem using deep image attributes. In this project, we adopt various insights from their work: in the preparation of the dataset, as well as focusing on relative virality rather than absolute virality.

Most recent works[13, 14] discuss the importance of identifying relative attributes in images, and propose a solution based on shallow image features and the RankSVM [11] formulation. These papers also discuss about utilizing pairwise(Siamese) networks to formulate relative virality based on the work of Souri et al [12] on 'deep relative attributes'.

My motivation, through this project, is to provide a pipeline for understanding the nature of content virality. I understand that virality is a function of both content and the nature of the network, however, I tackle the problem of understanding the information - specifically, visual and linguistic cues that can discriminate between viral and non-viral content. I have also tried to account for the disparity in the network topology present in the spread of viral information by creating a metric and dataset that balances the network effect.

Chapter 3

Datasets and Ground Truth Virality

3.1 Viral Posts Dataset

News handles on Twitter share 'formal' content belonging to fields that could be of interest to law enforcement agencies. To prepare the dataset, I selected 7 news handles namely: *'AJEnglish', 'cnni', 'Reuters', 'ani', 'ndtv', 'timesofindia',* and *'htTweets'*. These were chosen after analyzing the popularity of these handles, and inspecting their (diverse) content and (high) 'media posting frequency'.

I used the Twitter API to get all the posts that these handles posted in the early months of 2018. I was able to collect around 20k posts in the process. Many posts were discarded from this set to form a dataset of 17,022 posts last semester (discussed in the next section). I have filtered this dataset down to 15,098 posts this semester. The cleaning was done by taking into account the text in the post. If the caption did not contain any useful text (i.e. it only contained hyperlinks, stopwords or were fully empty etc.) they were not included in the dataset. Hence the dataset contains entries of images along with metadata like the text content of the post, number of likes, number of retweets, time (converted to GMT), and the Twitter handle. The metadata is utilized to calculate a 'Virality Score' for each post (as discussed in the next section). Fig. 3.1 shows a sample of the dataset.

3.2 Virality Score

The posts on Twitter are an expression of political, social and emotional views of the people sharing them. Each post can be liked and retweeted by a user. Viral content tends to be shared multiple times as it spreads across the network of users. Viral posts are thus the ones that have many likes, and have been retweeted often by different users. The

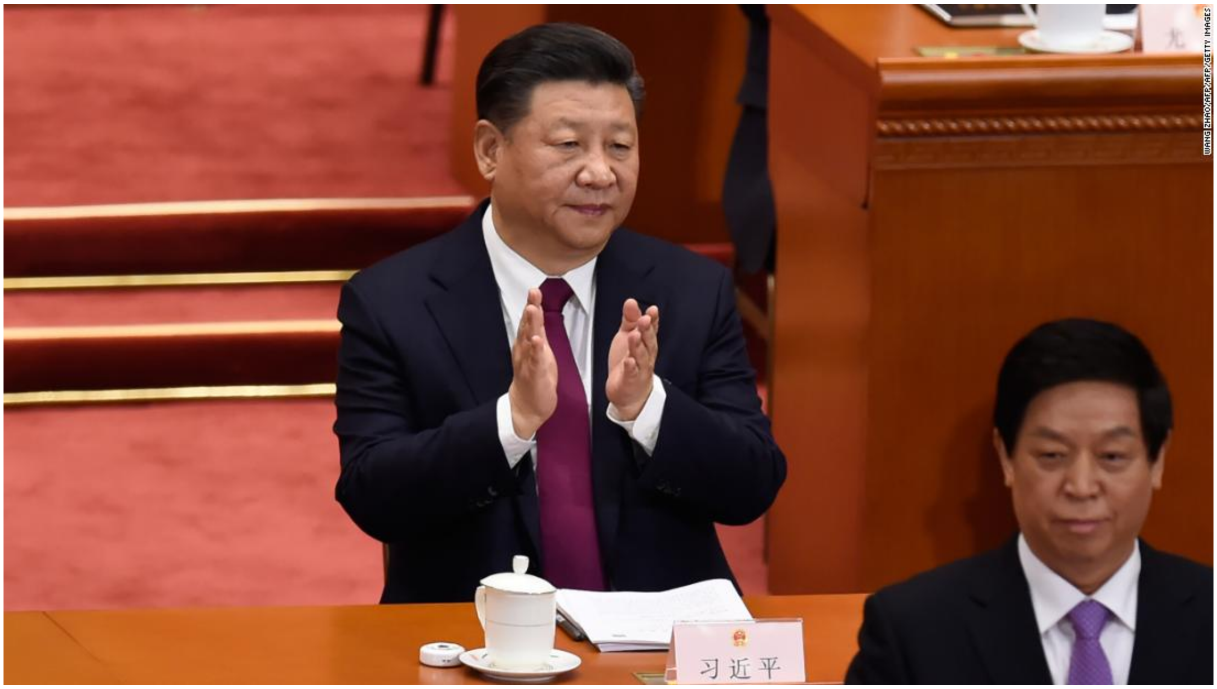

Figure 3.1: Metadata: [date: 2018/03/12, 11:09:03], [likes: 25], [retweets: 17], [text: President Xi Jinping's power knows no bounds]

latter is what differentiates virality from popularity. The formulation of virality score is inspired from Deza and Parikh [9].

Logically, the virality score should be a function of the likes (α) and retweets (β) on a picture. But, the number of likes and shares on a picture are affected by the person sharing it and the time when it is being shared. To cancel the effect of the page sharing the post and the time, we need to normalize the number of likes and retweets for each post. We divide the 24 hours of a day into 2 hour bins, such that, there are 12 bins on each day for each handle (i.e. 12x7 = 84 bins each day). Now each post is associated with exactly one bin depending on its handle and date. All the bins which contained less than 4 posts were deleted along with the posts in them. This was done to ensure the effectiveness of the normalization strategy: We calculate the mean likes ($\tilde{\alpha}$) and mean retweets ($\tilde{\beta}$) for each bin. To calculate the normalized likes and retweets for a post, we divide the absolute values (α, β) with the respective means ($\tilde{\alpha}, \tilde{\beta}$) of the associated bin. The 'Virality Score' of a post (θ) is given by the product of the normalized likes and the logarithm of normalized retweets (Eq. 3.1). It should be noted that this score will act as our ground truth for virality. Also, it was interesting to observe that, even in the same bin, the Virality Score varied a lot. This shows that the actual content of the post is an essential factor for virality.

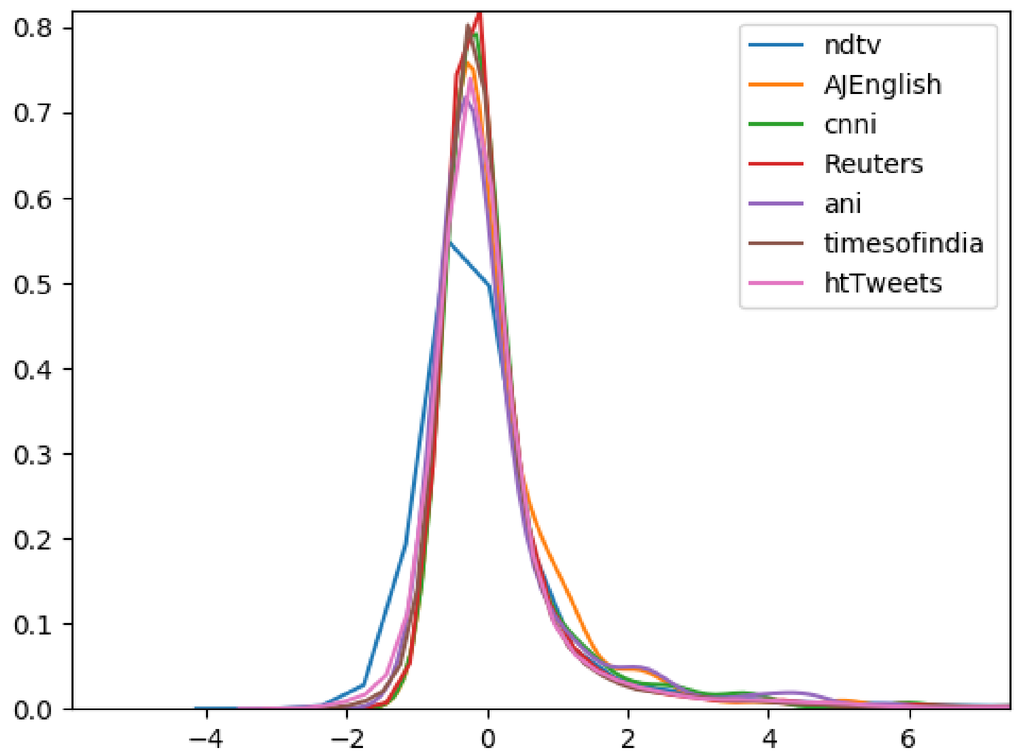

Figure 3.2: Normalized distribution of the formulated virality scores

$$\theta \ \textit{(Score)} \ = (\frac{\alpha}{\tilde{\alpha}}) \cdot \log(\frac{\beta}{\tilde{\beta}}) \tag{3.1}$$

Following is the distribution of the normalized scores for each twitter handle: We can see that the score on the filtered dataset gives more-or-less the same distribution for all the twitter handles, showing that the formula actually works well in providing an unbiased metric for defining virality.

3.3 Data Preparation

To prepare the data, I preprocessed each image and caption before sending it into the neural network (as discussed in next section). For each image, I normalized the value of each pixel to between 0 and 1 and resized it to 299x299x3. For each caption, I used regular expressions to clean it of the punctuations, URLs etc. Then the whole dataset is split into two parts with an 80:20 split for training and testing respectively. As we see in the next section, the model takes in a pair of posts as a time, these pairs have been made and used disjointly from training and testing sets respectively. It has been made sure that the model does not see an image from the test set at the time of training.

Chapter 4

Approach

Predicting the virality of indivual posts is a challenging task for both humans and machines. Therefore, predicting relative virality is a more logical and achievable task. That is, given a pair of posts, it easier to predict which of the two posts is more likely to be viral[19]. I used this measure of relative virality to predict absolute virality. For predicting relative virality, I used a Siamese Network as described below.

4.1 ViralityNet

The second model (which I used) is depicted in Fig. 4.1. This is based on the work of Souri et al [12] which utilizes a Siamese Network. During training, pairs of posts (P_i, P_j) are presented to the network, together with the target probability tij . Each post P is a combination of an image I and a caption T. If $P_i \geq P_j$ (post i is more viral than post j), then t_{ij} is expected to be larger than 0.5 depending on our confidence on the relative ordering of P_i and P_j . Similarly, if $P_i \leq P_j$, then t_{ij} is expected to be smaller than 0.5, and if it is desired that the two posts have the same rank, t_{ij} is expected to be 0.5. We chose t_{ij} from the set {0,1}, according to the available annotations in the dataset: we compare the virality scores of the two posts and assign a binary label to a pair of posts accordingly. The pair of posts then go though the feature learning and extraction part of the network (FeatureNet) (Fig. 4.2). This procedure maps the posts onto feature vectors ψ_i and ψ_j. Afterwards, these feature vectors go through the ranking layer. We choose the ranking layer to be a fully connected neural network layer with linear activation function, a single output neuron and weights w and b. It maps the feature vector ψ_i to the estimated absolute rank of that feature vector, $r_i \in R$, where

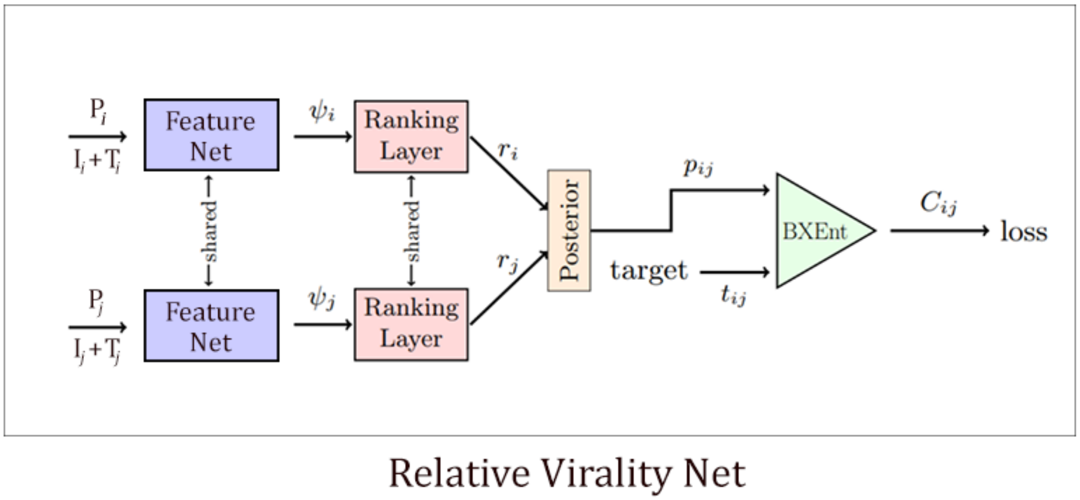

Figure 4.1: Siamese Network Architecture for Deep Relative Attributes

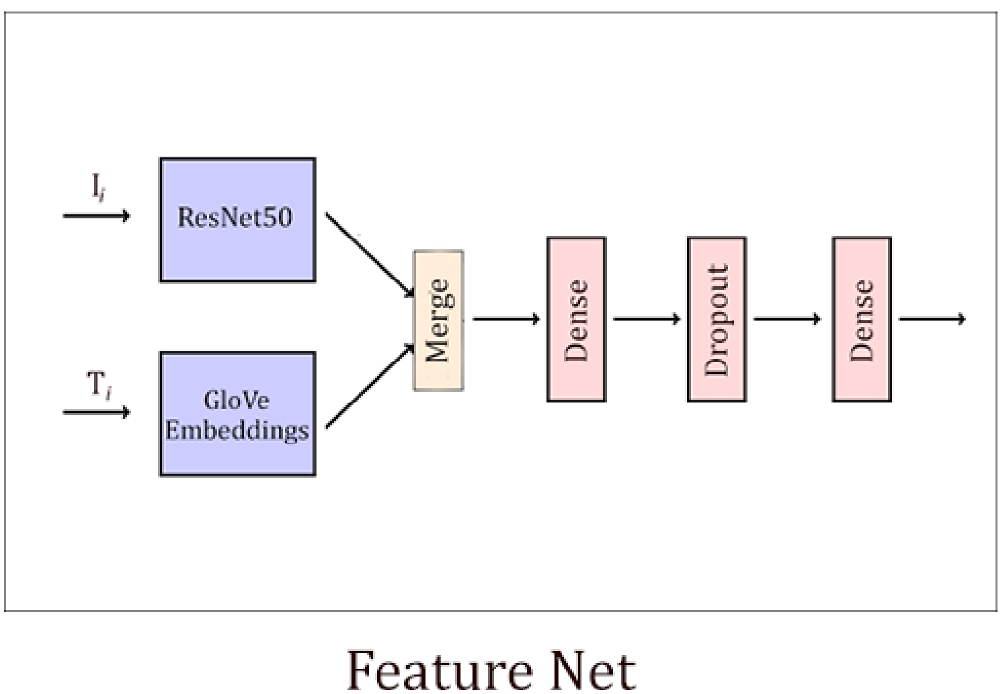

Figure 4.2: Feature Extraction Sub-Network

$$r_i := w^\intercal \psi_i + b \tag{4.1}$$

$$p_{ij} := \frac{1}{1 + e^{-(r_i - r_j)}} \tag{4.2}$$

$$C_{ij} := -t_{ij} \log(p_{ij}) - (1 - t_{ij}) \log(1 - p_{ij}) \tag{4.3}$$

The two estimated ranks r_i and r_j , for the two posts P_i and P_j in comparison, are then combined (using Eq 4.2) to output the estimated posterior probability p_{ij}=$P(P_i \geq P_j)$. This estimated posterior probability is used along with the target probability tij to calculate the loss, as in Eq. 4.3. This loss is then back-propagated through the network and is used to update the weights of the whole network, including both the weights of the feature learning and extraction sub-network and the ranking layer. During testing, we need to calculate the estimated absolute rank rk for each testing post Pk. Using these estimated absolute ranks, we can then easily infer both the relative or absolute attribute ordering(ranking), for all testing pairs.

4.2 FeatureNet

This sub-network works as a feature extractor for the input posts and returns a vector output to be fed into the ranking layer. The input image I is passed through a pre-trained(on ImageNet) model of ResNet50 to get a 2048 dimensional network and the input text T is converted to a 100 dimensional vector using GloVe embeddings pretrained on a Twitter dataset. The two vectors are merged to be passed into a dense layer of 200 dimensions. The output is passed through a dropout layer followed by another dense layer of 100 dimensions.This 100 dimensional output vector is then fed into the Ranking Layer in the ViralityNet.I tried a few models and the chose the one performing the best as the FeatureNet

Chapter 5

Results

I trained the network for 20,000 iterations with 40k pairs of posts in each iteration. These pairs were randomly chosen in each iteration from the train set. The model was tested on 40k randomly chosen pairs of posts from the test. It was ensured that, there was no repetition of 'pairs' in the testing and training sets by the very logic that was used to make them. Also, it was ensured that in each iteration, the number of positive and negative {0,1} samples were equal, to prevent any sort of bias toward one class.

5.1 Relative Virality

The metic for testing the model was taken as the mean 'accuracy' of prediction (which of the posts in the pair is more viral) over the test dataset.

The accuracy was calculated to be

$$Accuracy = 69.80\%$$

on the test dataset.

5.2 Absolute Virality

150 posts were carefully chosen from the Virality dataset that varied over Virality Score. To get the absolute virality for a test post, the post was fed into the proposed network one-by-one (paired up) with each of the 150 posts. The ratio of posts from which the test post was rated to be more viral was taken as the absolute virality score of the post on a scale of 0-100. Basically, the average of the confidence scores for the 150 pairs is the final

predicted virality score. It was interesting to note that, the model predicted images that contained ‘people’ to be more viral.

Bibliography

[1] J. Leskovec, L. A. Adamic, and B. A. Huberman. *The dynamics of viral marketing.*. Transactions on the Web, 2007. 1, 2, 4

[2] A.-L. Barabasi. *The origin of bursts and heavy tails in human dynamics.* Nature, 2005. 1

[3] P. Shakarian, S. Eyre, and D. Paulo. *A scalable heuristic for viral marketing under the tipping model*, 2013. 1, 2

[4] Lichan Hong, Gregorio Convertino, and Ed H Chi. 2011. *Language Matters In Twitter: A Large Scale Study*

[5] Sasa Petrovic, Miles Osborne, and Victor Lavrenko. 2011. *RT to Win! Predicting Message Propagation in Twitter*

[6] Amandianeze O Nwana, Salman Avestimehr, and Tsuhan Chen. 2013. *A latent social approach to youtube popularity prediction.* In Global Communications Conference (GLOBECOM), 2013 IEEE. IEEE, 31383144.

[7] Henrique Pinto, Jussara M Almeida, and Marcos A Gonalves. 2013. *Using early view patterns to predict the popularity of youtube videos.* In Proceedings of the sixth ACM international conference on Web search and data mining. ACM, 365374.

[8] David A Shamma, Jude Yew, Lyndon Kennedy, and Elizabeth F Churchill. 2011. *Viral Actions: Predicting Video View Counts Using Synchronous Sharing Behaviors.*. In ICWSM.

[9] Arturo Deza and Devi Parikh. 2015. *Understanding image virality. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 18181826.*

[10] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. 2014. *What makes an image popular?.* In Proceedings of the 23rd international conference on World wide web. ACM, 867876.

[11] M. Spain and P. Perona. *Measuring and predicting object importance.* IJCV, 2011. 2

[12] Yaser Souri, Erfan Noury, and Ehsan Adeli-Mosabbeb. 2015. *Deep Relative Attributes.* arXiv preprint arXiv:1512.04103 (2015).

[13] Abhimanyu Dubey, Sumeet Agarwal. 2017. *Modeling Image Virality with Pairwise Spatial Transformer Networks*

[14] Xavier Alameda-Pineda, Andrea Pilzer, Dan Xu, Nicu Sebe, Elisa Ricci. 2017. *Viraliency: Pooling Local Virality*

[15] Chris Burges cburges@microsoft.com,Erin Renshaw. 2015. *Learning to Rank using Gradient Descent*

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun. 2015. *Deep Residual Learning for Image Recognition*