Picture or Words: Predicting Twitter Image Post Popularity with Deep Learning

Yihong ZHANG†, Adam JATOWT†, and Yukiko KAWAI††

† Kyoto University
†† Kyoto Sangyo University
E-mail: †yihong.zhang.8z@kyoto-u.ac.jp, ††adam@dl.kuis.kyoto-u.ac.jp, †††kawai@cc.kyoto-su.ac.jp

Abstract Predicting popularity of a post in microblogging services such as Twitter is an important task that is beneficial for both publishers and regulators. Traditionally, the prediction is done through various manually designed features extracted from post and user contexts. In recent years, deep learning models such as convolutional neural network (CNN) have shown significant effectiveness in image processing. In this paper, we make a novel investigation of the effectiveness of deep learning models in predicting image post popularity, with the raw image as the input. In contrast to previous works that use existing model trained for object detection, we trained a CNN model targeting directly at predicting popularity. We show that dedicated CNN is more effective than networks trained for other purposes.

Key words popularity prediction, microblog, deep learning

1. Introduction

A social media such as Twitter that has hundreds of millions monthly active users is nowadays an important platform for information sharing. On Twitter, in addition to personal users who post information about their daily lives, there are also companies who promote their products and organizations that make announcements and advertisements. It would be of great interest for these publishers to know the future popularity of their posts. In this paper, we deal with the problem of predicting tweet popularity *before* posting the tweet. This is in contrast to some existing works that predict tweet popularity *after* posting the tweet, for example, based on early propagations [3]. Particularly, we aim to predict the popularity of tweets with images, as many advertisements on Twitter are based on the content of the image and some contextual information. Such before-hand prediction can have many benefits, such as allowing publisher to adjust their tweet content in order to get higher popularity.

On Twitter, there are two common measurements of tweet popularity, namely, the number of retweets and the number of likes. Retweeting is the activity to re-post someone else's tweet in the retweeting user's account, while liking is the activity to click a button on the tweet to indicate admiration, without repeating the tweet. The count of both activities received can indicate the popularity of the tweet. However, liking tends to indicate that the tweet is sentimentally admirable, and retweeting often indicates tweet containing important information, regardless of its sentiment value. In this paper, we use both measurements.

While the reason for a tweet to receive likes may be considered rather simple, the reason to receive retweets is more complex. There were cases that a tweet from some obscure personal account suddenly went viral because it had been retweeted by certain celebrity. Such posterior factors are however not in the scope of this paper. We rather focus on the information one can get prior to posting the tweet. Specifically, we identify two pieces of information that is critical for predicting popularity in addition to the actual content of an image, namely, the number of followers and the time elapsed in hours. On Twitter, a tweet posted by a user is usually automatically displayed in the pages of all the followers of that user, so the number of followers means the number of initial audience of the tweet, a proportion of which will then retweet or like the tweet. Therefore the number of followers is critical information. Also getting retweets and likes is an accumulation process, therefore the time elapsed since posting is also important. Our experimental analysis shows that, without inputting these information, one cannot get meaningful predictions. In an example application, a user would provide the image and the text message, and specify the number of followers and time elapsed, and she can get a prediction of the number of retweets and likes the tweet is likely to receive once posted.

For analyzing images we will use the latest findings in deep learning. It has been shown that convolutional neural network (CNN) is particularly effective in processing images, with its ability to capture local features of the image [6]. However, existing works on predicting image popularity mostly use pre-trained network targeting object recognition. In contrast, in this paper, we propose a CNN specifically trained for predicting popularity. We extend the standard CNN with additional inputs into one of its middle layer. In the experimental analysis we will demonstrate that this dedicated network can achieve higher prediction accuracy than pre-trained networks. Another question we are interested in finding out is whether the image contains more predictive information than the text in the same tweet. We therefore compare image-based prediction with text-based prediction and show results. In Section 3, we will describe the proposed CNN for popularity prediction, and in Section 4, we will present our experimental analysis.

2. Related Works

Treated either as a classification problem [8], [12], [14] or a regression problem [2], [4], [5], predicting image post popularity in existing works is mostly done through supervised learning with manually designed features. These features can be grouped into image features, text features, and context features. Early works using images features focus on the low-level image aspects [8], [14]. Totti et al. propose a feature set that includes low-level image information such as color channel statistics, dominant colors, contrasts, and focus [14]. They also consider social context such as user gender, number of followers, as well as temporal information such as the day of the week for the post. They find that social features are more effective than meta image information. McParlane et al. too consider image color, while providing more advanced features such as number of faces detected and scenery information of the image, for example indoor or outdoor [8]. They also include text features such as tf-idf of image tags. Their findings show that text features are much more effective than any of the image-based features.

In recent years, neural network models for image processing have become popular, and has been adopted in image post prediction studies. Khosla et al. propose a feature set that includes output of an existing neural network trained to detect objects in the image, in addition to low-level image features such as color and texture [5]. They find that the neural network-based features achieve better results than any other features. Gelli et al. also use a neural network trained for object detection, as well as a neural network trained for sentiment detection, to generate their feature set [4]. They also tested text-based features such as BOW and recognized name entities. They found that image-based features can achieve a spearman correlation of 0.36, while for text-based features the correlation can reach 0.63. However, although the user-chosen image tags in the dataset they use contain important predictive information, such tags are not available in Twitter. Cappallo et al. propose a ranking method for predicting image popularity, also using pre-trained network for object detection [2]. They show that by dividing images into popular and unpopular categories, the prediction accuracy can be improved. The authors also claim their result compares favorably to [5]. However, only some parts of features in the compared work are used.

Traditionally, text data are represented as BOW vectors. Recently, neural network model for text processing has also been adopted in post popularity prediction. For example, Ramisa et al. propose to use pre-trained word2vec [7] word embedding model to convert post text into vector, before it can be applied to tasks such as post popularity prediction [10]. They found that for $l1$ norm BOW and tf-idf perform better than word embedding, but for $l2$ norm the word embedding performs better. Almgren et al. [1] also use pre-trained word2vec model, in addition to BOW and clustering to generate text vector. They show that their approach produced better results than [8]. Stokowiec et al. use word embedding trained on Wikipedia and Gigaword, publicly available as GloVe [9], and compare them with SVM predictor and BOW representation [12].

However, there is still lack of work aiming at performance comparison between neural network representation of text and image with respect to popularity prediction. In this paper we fill in this gap.

3. Hybrid CNN for Predicting Popularity

We design a deep learning model that harnesses the power of convolutional networks to comprehend local image features, and at the same time it allows an additional information input to be used for training alongside the images. The resulting model is a combination of a convolutional network and a fully-connected network. Figure 1 illustrates the structure of our proposed model.

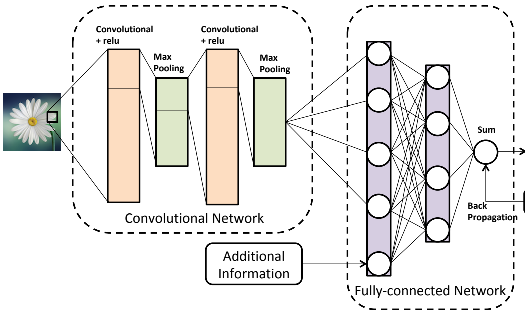

Figure 1 Structure of the proposed hybrid CNN model. Note that only selected network nodes are shown.

3. 1 Network Structure

In the convolutional network, we setup four layers. The first and third layers are convolutional layers and the second and forth layers are max-pooling layers. For a convolutional layer, we use a number of kernels that extracts local features from the image. The output of each convolutional layer is then *max-pooled* in the max-pooling layer. We use rectified linear units (relu) as the activation function for the convolution layer, because it is efficient for a large network and can avoid vanishing or exploding gradient problems in the training phase. This is a well-known structure of convolutional network built for image processing tasks, appearing in several previous works [6]. We skip specifying details such as the number of nodes until the next section.

In the fully-connected network, we setup two layers, each containing a number of fully-connected neurons. The inputs to the first layer are the outputs of the second max-pooling layer *and* additional information. As we discussed in the introduction, we use two pieces of additional information, namely, number of followers and hours elapsed. Thus the number of inputs to the first fully-connected layer is the number of outputs of the second max-pooling layer +2. The image features and the contextual information are thus combined in a single network. Finally the outputs of the second fully-connected layer are aggregated as sum in the final node, which produces the prediction value.

When training the network, we use mean square error (MSE) of the predicted value and the actual popularity as the cost function. A gradient-based optimizer then iteratively improves network weights using back-propagation.

3. 2 Network Implementation and Training

We implement the proposed network using Google Tensorflow[?i??1?j], which provides a number of interfaces for quick construction of deep neural networks. First of all, we resize all input images to 64×64 pixels, and input them as vectors representing pixel values. After trying a number of different values, we settle on the following model parameters. For the first convolution layer, we use 32 5×5 kernels. For the second convolution layer, we use 64 5×5 kernels. For both max-pooling layers, we use pool size of 2×2 with strides of 2. As the result, there are 16,384 outputs from the second max-pooling layer. For the first fully-connected layer, which takes 16,386 inputs, we set 100 neurons. We setup 60 neurons for the second fully-connected layer. Note that this setup is not necessarily optimal, and we have found different parameter values produce similar results.

We use the Adam (adaptive moment estimation) optimizer implemented in Tensorflow. This optimizer is a variety of stochastic gradient descent that uses adaptive learning rates, and has proven effective in providing optimal results faster. We set the initial learning rate as 0.001. In the experiments, we run 1000 training epochs. The cost function generally converges during the training. The trained model is then applied to the test data for evaluation.

4. Experimental Analysis

We conduct experiments on image tweets dataset to test the effectiveness of our approach. Particularly, we are interested in finding out whether our dedicated hybrid CNN model can outperform an object recognition based network and text-based prediction model. In this section, we will describe our data collection process and baseline methods, before discussing the evaluation results.

4. 1 Data Collection

We collect a number of tweets with images using Twitter's Sample API[?i??2?j]. The Sample API returns a small random sample of all public tweets posted in realtime. For our study, we are interested in those tweets that have accumulated a certain amount of popularity over a period of time. Therefore we select from sampled tweets those that contain images and have already accumulated more than 100 retweets[?i??3?j]. We collected in this way 107,558 tweets. We also recorded the time of the collection and removed tweets with less than seven days elapsed between the posting time and the time of data collection. Furthermore we eliminated outliers as follows. Specifically, we removed tweets that have more than 10,000 retweets or 10,000 likes, because higher popularity may indicate phenomenon of going viral, which is still difficult to be explained. We also removed users who have more than 100,000 followers, since it has been shown that the celebrity status of a poster can give the tweet unusually high popularity regardless of its actual content [15]. Finally, we have 33,558 tweets that satisfy all the filtering requirements. Figure 2 shows the distribution of the number of retweets and likes in the dataset. We can see that the majority of both measurements are less than 2,000. The Pearson correlation between the number of retweets and likes is 0.631, which is not a very strong correlation. Thus predicting one measure of popularity does not necessary predict the other.

The numbers of retweets, likes, followers of the user, and hour elapsed since posting are all normalized by their maximum values, which are 9,989, 9,998, 99,891 and 52,155, respectively. For experiments, we divide the data into two equal parts as training data and testing data following the approach of [5].

?i??1?jFhttps://www.tensorflow.org/

?i??2?jFhttps://developer.twitter.com/en/docs/tweets/sample-realtime/overview/GE

?i??3?jFWe can trace back the original tweet from the retweets

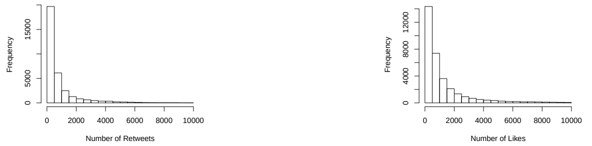

Figure 2 Distribution of the number of retweets and likes in the dataset

4. 2 Baseline Methods

We compare our method against previous popularity prediction methods based on image and text. Particularly, we focus on those methods that involve deep leaning.

Inception-SVR. A common approach of the previous popularity prediction works is to use an existing network trained for object recognition, and convert images into vectors by extracting the output of the final neural layer, before the classification output. The vectors extracted in this way essentially represent the semantics of the images. Extracted vectors are then processed by standard regressors such as Support Vector Regression (SVR) to produce predictions. It has been shown that this approach is more effective than methods depending on the low-level image features such as color and texture [5]. Among the existing works, the most common pre-trained network used is the “AlexNet” trained on 1.3 million images, the winner the 2012 ImageNet challenge [5], [6]. In recent years, more effective networks have been proposed. For instance, Szegedy et al. have release a network called Inception [13]. It is deeper and wider than AlexNet, and involves asymmetric convolutions. The latest version, Inception-v3, has 42 deep layers, and is publicly available[?i??4?j]. In the experimental evaluation for object classification, Inception-v3 reaches a top-5 error rate of 3.46%, compared to 15.3% reached by AlexNet, and 6.67% reached by the original Inception network. Therefore in this paper, we choose Inception-v3 instead of AlexNet as the example of pre-trained network. To use the Inception-v3 Network, we add short code to the Inception-v3 program to extract the output of the third pooling layer, which is a vector of 2,048 length representing the semantics of the input image. We add to this vector the two contextual information signals, i.e., the number of followers and the number of hours elapsed, and run it with SVR to train a model for generating popularity prediction outputs.

BOW. Our first text based baseline is Bag-of-Words (BOW). This is a commonly used baseline for text-based analysis [11], [12]. For this and the next baselines, we consider the tweet text in our dataset. We first generate a dictionary by selecting words that appear more than 100 times in the dataset. We get a vocabulary size of 278. Because tweets are short, we use binary BOW for tweet representation. Specifically, for each tweet, we generate a vector $\{w_1, ..., w_{278}\}$ where $w_i \in \{0, 1\}$ indicates whether the i-th word in the dictionary appears in the tweet. After adding the number of followers and count of elapsed hours, we run SVR for training and prediction.

Table 1 Evaluation Results

		hybrid CNN	Inception + SVR	BOW	GloVe
retweets	median err	0.074	0.041	0.089	**0.036**
	spearman ρ	**0.267**	0.206	0.233	0.262
likes	median err	0.086	0.057	0.082	**0.051**
	spearman ρ	0.347	0.315	0.332	**0.367**

GloVe. In recent years, word embedding has become a popular method for text-based analysis. Techniques such as Word2Vec create distributed representation of words through continuous bag-of-words or skip-gram. Such representation captures the context and by this represents the semantics of words. Previous works have shown the effectiveness of using pre-trained word embedding models for classification and regression tasks [1], [10]. For our study, we use an approach similar to the method proposed in [10], which uses mean value of word embedding for each word in the tweet. Note that we found CNN with padding proposed in the same paper to produce poorer results, thus we decided to not include it in the comparison. Similar to [12], we use a pre-trained word vector representation called GloVe T100 [9], which is trained on two billion tweets. This model contains 1.2 million large vocabulary, and has a vector length of 100. We generate a vector for each tweet using the mean vector of the words in the tweet, and we run SVR for training and prediction after having added the number of followers and count of hours elapsed.

4. 3 Evaluation

We use two measurements to evaluate prediction accuracy, median error and spearman correlation. Median error is taken as the median of absolute error between predictions and the true popularity values. We use median error because it is more stable given the large variation in tweet popularities. Spearman correlation is based on the correlation of the ranking of prediction and true values, and reflects relative popularity that is less influenced by the variance. Both measurements have been used in the previous studies of popularity prediction [5], [10]. The evaluation results are shown in Table 1. The first column lists the result of the proposed hybrid CNN method.

First, we compare our dedicated network to pre-trained

[?i??4?j] https://www.tensorflow.org/tutorials/image_recognition network. For both retweets and likes, the dedicated hybrid CNN reaches higher spearman correlation value than Inception network. However, the mean absolute error is smaller for Inception network, because it is a larger network and is more stable. Then we compare dedicated network to text-based methods. We can see that hybrid CNN method reaches higher spearman correlation value in both measurements than simple BOW method, and in the case of retweets it also has lower median error. However, GloVe word embedding method performs better than the hybrid CNN method, having lower error in both the measurements, and achieving higher spearman correlation value for likes. The hybrid CNN method nevertheless reaches the highest spearman correlation for retweets among all methods. To conclude, the dedicated network can obtain higher accuracy in predicting relative popularity than the pre-trained network. It also makes better prediction than a simple text-based method such as BOW. However, more advanced text-based methods such as word embedding can capture better prediction signals from text and can make better prediction than the image-based network. This finding is consistent with previous work [4], even though text in tweets has different purpose than tags of Flickr images.

We also notice that the relative popularity in terms of retweet number is more difficult to predict than likes. This is reasonable because, as explained in the introduction, the act of retweeting changes the audience of the tweet. The factor of a tweet receiving many retweets, or *going viral*, is more difficult to explain and predict than a tweet receiving many likes, which is mostly based on the tweet content itself. Nevertheless, our hybrid CNN is able to capture the relative popularity as retweet number better than other image or text-based methods.

5. Conclusion

In this paper, we study predicting image tweet popularity based on image and text contents. We propose a dedicated hybrid convolutional neural network that captures image local features with regard to popularity measurements. We compare our dedicated network to a pre-trained network built for object detection and text-based methods. We find that our dedicated network is able to make better prediction than pre-trained network, and is comparable with state-of-art text-based methods. In future works, we plan to further investigate the factors that produce popularity signals in image and text contents.

6. Acknowledgments

This research and development work was supported by the MIC/SCOPE #171507010.

References

[1] K. Almgren, J. Lee, et al. Predicting the future popularity of images on social networks. In *Proceedings of the The 3rd Multidisciplinary International Social Networks Conference on SocialInformatics 2016, Data Science 2016*, page 15. ACM, 2016.

[2] S. Cappallo, T. Mensink, and C. G. Snoek. Latent factors of visual popularity prediction. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 195–202. ACM, 2015.

[3] S. Gao, J. Ma, and Z. Chen. Modeling and predicting retweeting dynamics on microblogging platforms. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 107–116. ACM, 2015.

[4] F. Gelli, T. Uricchio, M. Bertini, A. Del Bimbo, and S.-F. Chang. Image popularity prediction in social media using sentiment and context features. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 907–910. ACM, 2015.

[5] A. Khosla, A. Das Sarma, and R. Hamid. What makes an image popular? In *Proceedings of the 23rd International Conference on World Wide Web*, pages 867–876. ACM, 2014.

[6] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[7] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1188–1196, 2014.

[8] P. J. McParlane, Y. Moshfeghi, and J. M. Jose. Nobody comes here anymore, it's too crowded; predicting image popularity on flickr. In *Proceedings of International Conference on Multimedia Retrieval*, page 385. ACM, 2014.

[9] J. Pennington, R. Socher, and C. 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.

[10] A. Ramisa, F. Yan, F. Moreno-Noguer, and K. Mikolajczyk. Breakingnews: Article annotation by image and text processing. *arXiv preprint arXiv:1603.07141*, 2016.

[11] B. Sriram, D. Fuhry, E. Demir, H. Ferhatosmanoglu, and M. Demirbas. Short text classification in Twitter to improve information filtering. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 841–842, 2010.

[12] W. Stokowiec, T. Trzciński, K. Wołk, K. Marasek, and P. Rokita. Shallow reading with deep learning: Predicting popularity of online content using only its title. In *International Symposium on Methodologies for Intelligent Systems*, pages 136–145. Springer, 2017.

[13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[14] L. C. Totti, F. A. Costa, S. Avila, E. Valle, W. Meira Jr, and V. Almeida. The impact of visual attributes on online image diffusion. In *Proceedings of the 2014 ACM conference on Web science*, pages 42–51. ACM, 2014.

[15] S. Wu, J. M. Hofman, W. A. Mason, and D. J. Watts. Who says what to whom on Twitter. In *Proceedings of the 20th International World Wide Web Conference*, pages 705–714, 2011.