Retweet Wars: Tweet Popularity Prediction via Dynamic Multimodal Regression

Ke Wang Mohit Bansal Jan-Michael Frahm
Department of Computer Science, University of North Carolina Chapel Hill
{kewang,mbansal,jmf}@cs.unc.edu

Abstract

If a picture is worth a thousand words, then images should be utilized together with other available data modalities when predicting the virality of online posts, such as tweets. In this paper, we re-visit the tweet popularity prediction problem by considering all data modalities: tweet language semantics, embedded images, author' social relationships, and the diffusion process of tweets. To model the content of tweets, we propose a joint-embedding neural network that combines visual, textual, and social cues together. Such content features can be either used for prediction directly, or for pre-conditioning a 'dynamics RNN', which models the message propagation process. A novel Poisson regression loss is optimized to train the network. We demonstrate that content based features can be used to improve upon social features and dynamics features via our joint-embedding regression model. Our model outperforms the state-of-the-art on multiple large-scale real-world datasets collected from Twitter.

1. Introduction

The world is better connected than ever before. On social networks, users are connected to every other user by an average separation of 3.57 [1]. Short communication distance and ease of access make online social media an increasingly popular venue for information sharing. However, convenience comes with a cost. Both individuals and organizations can be easily overwhelmed by the sheer volume of online posts or misled by wide-spread rumors. Therefore, the ability to predict which post has a high popularity potential in its early stage can help individuals improve their communication efficiency and also allow organizations sufficient time for remedial actions. Reliable forecasting of online content popularity is thus a vital need.

Popularity prediction has long interested various research communities [34]. Previous methods approach this problem by analyzing the author's influence on the social network [40], or the early dismantling behaviors of posts [41]. Social features and cascading process modeling are the dominant foundations for popularity predictions. Recently, deep learning based methods have revolutionized many vision and language tasks, providing new ways to analyze visual and textual content in online posts. Hence, the question arises whether such visual and textual content can help improve upon the popularity prediction accuracy? In this paper, we study the role of content for the popularity prediction task in both the *static* setting, where only the post is known, and the *dynamic* scenario, where the early retweeting process is also known. We found that by carefully blending the different content modalities together, improvement can be brought to the virality prediction task. Subsequently combining jointly embedded content features with social cues and temporal cascading processes, we show that, in addition to *who you are*, *what you say* and *what you show* are also important indicators of the breadth of message's reach.

We present the virality prediction problem on the Twitter domain. For the static prediction scenario, we propose a multimodal regression model that jointly considers the vi-

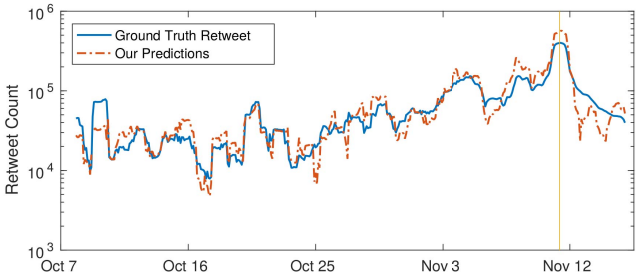

Figure 1: We demonstrate our method on the 2016 US presidential election tweets. We collected tweets containing relevant keywords (*"president", "vote", "election", "Clinton", "Trump"*) from October 8, 2016 to November 14, 2016. Retweets (solid blue line) on the presidential campaign were exponentially increasing before the election day (the orange vertical line). Our proposed method (dotted yellow line) accurately predicted such a trend. For visualization, the tweets are grouped into bins of one hour width based on their post time.

[1]Three degrees of separation: https://research.fb.com/three-and-a-half-degrees-of-separation/

978-1-5386-4886-5/18/$31.00 ©2018 IEEE
DOI 10.1109/WACV.2018.00204

sual and textual data as well as the authors' social features to predict the potential influence of tweets, as measured by their retweet counts. We adopt the Inception-ResNet CNNs [36] and LSTM-RNNs [16] to model the visual and textual features, respectively. We use in-domain word embeddings specifically trained on tweet-style language as input to the LSTM-RNNs. We also explicitly model the shared semantic relationships between the tweet text and embedded images using a joint embedding model trained under a bidirectional ranking loss. Deeply learned features, together with user-specific social features, are then used to learn a Poisson regression model which predicts the potential influence of the given tweet. We propose to use recurrent neural networks to predict the final retweet count for a given dynamic sequence of retweeting actions. The static content based features are used as pre-conditioning for the recurrent network.

To evaluate our proposed model, we use both existing as well as our novel large-scale multimodal Twitter datasets. We collect and present two datasets: one from the year 2015 with 14 million tweets containing over 3 million images, and one from the next year (2016) with 10 million tweets containing 2 million images. The latter is used to test the generalization of our methods. On both our datasets and on the existing MBI-1M dataset [6], our method outperforms state-of-the-art multimodal methods. We also assembled a temporal dataset that records the propagation process of tweets, which we used to study the performance of our dynamic prediction models.

To summarize, our main contributions are:

1. A multi-modal neural network model that harnesses all available Twitter data modalities: visual, textual, social, and temporal cues;
2. A joint embedding model, trained under bidirectional ranking constraints, that explicitly captures the shared semantic relationships between visual and textual data;
3. A novel Poisson regression model for predicting retweet count based on all available data modalities;
4. Demonstrated the role of content for popularity prediction in both static and dynamic scenarios;
5. Ablation and attribute analysis to explain model component and modality contributions, as well as what visual and textual features the model is learning.

The remainder of this paper is structured as follows. Section 2 briefly reviews the related literature. We describe the problem formulation and our network architecture in Section 3. Section 4 covers experimental results and discussion. Finally, Section 5 concludes the paper.

2. Related Work

Our work studies the problem of tweet popularity predictions. We draw inspirations from multiple disciplines.

Social networks Compared with other social media, Twitter has particularly distinctive features. As pointed out in [18] and [22], Twitter is not only a social network but also a news medium. Information spreads on Twitter at astonishing speeds, providing the possibility for event detection [5], sentiment classification [14], popularity prediction [6], and tweet-based language processing [10]. We not only train a Twitter-specific word embedding and language model to learn the Twitter language, but also fine-tune pre-trained CNN models on Twitter images.

Content-based popularity prediction Popularity prediction for online social networks is a fairly well-studied problem. Content based prediction infers the popularity using textual and/or visual features. For example, [25] utilized textual, visual, and social cues to predict the image popularity on Flickr. [19] used contextual and deeply-learned visual features to explore the factors influencing an online photo's popularity. [40] combined visual, textual, and social features to predict popularity in the fashion domain but only use tag-based text and no joint embedding models. [11] showed that mid-level image features trained on deep networks improved the performance of image virality prediction. [37] showed that carefully crafted wording of the message can help propagate the tweets better. Although some of the previous works incorporate multimodal information, only simple direct feature fusion is used [19, 40], whereas our work explicitly exploits the inter-domain relationships via joint embedding. We find that this joint embedding approach is crucial to achieve complementary performance improvements.

Diffusion-based popularity prediction A complementary line of popularity prediction methods do not rely on the content but instead use social features such as user influences, combined with real-time monitoring of the diffusion process to make predictions. [17] showed social-oriented features were the best performer to predict image popularity on Twitter. [42] utilized image features extracted from CNNs and social-oriented features for popularity prediction. [1] used temporal evolution patterns to predict the popularity of online user-generated content. [7] used temporal and structural features to predict cascades of photo shares on Facebook. [41] model the retweeting cascades as a self-exciting point process. Similarity, our work also uses a recurrent neural network to model the temporal diffusion of the retweet process. In contrast to the above, our dynamics RNN is explicitly pre-conditioned on the content features and the social features.

Deep learning Deep neural networks empower computational models to learn rich feature representations at multiple levels of abstraction. Computer vision has benefited greatly from convolutional neural networks (CNNs), for classification [15], semantic segmentation [29], and object detection [23]. Deep-learning-based methods have also influenced natural language processing (NLP), from word

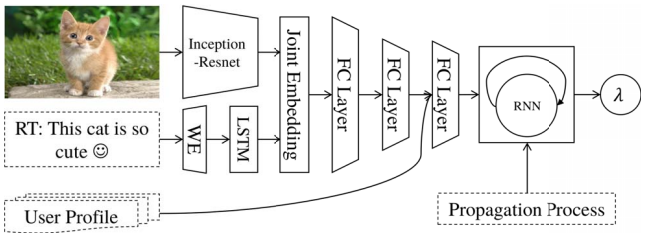

Figure 2: Our proposed multi-modal model to predict tweet popularity. A state-of-the-art Inception-Resnet CNN model is used to extract visual features and an LSTM is used to extract textural features. Visual and textual representations are then mapped to a common space by a joint embedding network. For static scenario, the joint content feature together with social cues are used as input to the Poisson regression model. For dynamic settings, jointly embedded content features and social features are used to precondition the dynamics RNN, which predicts the Poisson model by looking at early stage propagation data.

embeddings [26] and language modeling with recurrent neural networks (RNNs) [27] to syntactic parsing [32] and machine translation [8, 35]. Our work is built upon state-of-the-art CNN networks to extract rich visual features for Twitter-style images, and LSTM-RNN models to extract Twitter-style language semantics.

Multimodal deep learning Multimodal machine learning integrates and models multiple communicative modalities, such as linguistic, acoustic and visual messages. For example, [28] used deep autoencoder models to learn multimodal features for audio-visual speech classification tasks. [33] propose to use deep Boltzmann machines to learn generative models from multimodal data. Recent advances in computer vision and natural language processing have piqued a common interest in applications connecting visual information and textual descriptions, such as image captioning [31, 38] and visual question answering [3, 13]. [2] proposed a deep learning based extension to canonical correlation analysis (CCA). [12] used ranking losses to learn the linear transformations on visual and textual features. In our work, we follow the lead of [39] in using a bi-directional ranking loss to learn non-linear transformations that correlate tweet text and images such that they are in a joint, shared space and allow easier feature learning for the regression model, leading to stronger improvements for our task.

3. Methodology

We consider the problem of predicting tweet popularity, that is the number of times a tweet will be retweeted. A tweet T containing an image I and language descriptions L is first issued by its author U. At time t_i the tweet of interest accumulates r_i retweets. Such dismantling process is recorded as $D = \{(t_0, r_0), (t_1, r_1), \ldots, (t_N, r_N)\}$. Note that D may only record the early stage of the dismantling process. The maximum retweet count during the data collection period is used as the ground-truth retweet count r_{gt}. As a discrete probability model, the Poisson distribution characterizes the probability of a given number of events occurring during some time period. Therefore, the retweet count r of a tweet $T(I, L, U, D)$ follows a Poisson distribution:

$$P(R = r|\lambda) = \frac{e^{-\lambda}\lambda^{-r}}{r!} \tag{1}$$

where the latent variable $\lambda \in \mathbb{R}^+$ defines the mean and variance of the underlying Poisson distribution. For a *static* scenario, where the propagation information D is not available, the dynamics RNN module in Figure 2 is removed and only (I, L, U) are used in the Poisson regression model.

We propose a neural network model that directly maximizes the probability of the retweet count r given the tweet information (namely the image I, the tweet text L, the user profile U, and the early stage propagation information D):

$$\theta^* = \arg\max_{\theta} \prod_{(T,r)} P(R = r|T; \theta) \tag{2}$$

where θ are the neural network parameters. Our proposed network combines multi-modal information from the unseen tweet $\tilde{T}$ to predict the Poisson parameter $\tilde{\lambda}$ for its latent Poisson distribution $P(R)$. The retweet count prediction $\tilde{r}$ for $\tilde{t}$ can then be easily inferred by maximizing $P(R; \tilde{\lambda})$:

$$\tilde{r} = \max\left(\left\lceil\tilde{\lambda}\right\rceil - 1, \left\lfloor\tilde{\lambda}\right\rfloor\right) \tag{3}$$

3.1. Multimodal Feature Network

Figure 2 gives an overview of our overall network architecture. Our proposed model consists of two stages: a feature extraction network and a dynamic RNN network. The feature extraction network processes image I, language L, as well as user profile U. The output features are used to pre-condition the dynamic RNN network. The preconditioned dynamics RNN then processes the propagation process data D to estimate the hidden Poisson parameter λ.

In the feature network, a convolutional neural network (CNN) transforms I into a fixed length feature vector $f_{CNN}(I)$. A long short-term memory recurrent neural network together with tweet-trained word embeddings encodes the variable length tweet language L into a fixed dimensional feature vector $f_{LSTM}(L)$. We employ an extra joint embedding network to map the different modality features $f_{CNN}(I)$ and $f_{LSTM}(L)$ into a common space.

Visual CNN We adopted the state-of-the-art Inception-Resnet architecture [36] to extract a rich feature representation from the Twitter images I. We chose such architecture because: 1) the Inception-Resnet architecture can produce high quality visual features from images, as shown by its leading performance on ImageNet challenge [9]; 2) using

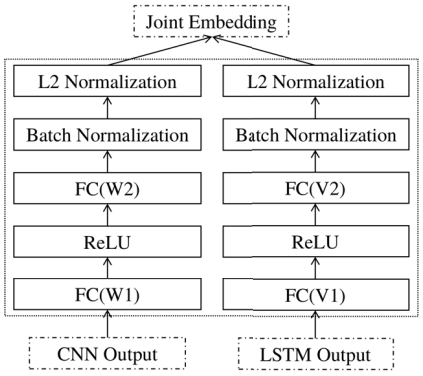

Figure 3: Joint embedding network: the two branches of the network do not share weights. CNN output f_{CNN} and LSTM output f_{LSTM} are fed into separate branches. The output of the two branches is L_2-normalized and have the same dimensions.

Table 1: Tweet text pre-processing rules.

Category	Before	After
URL	t.co/abc	abc.xyz
Hashtag	#love	<hashtag> love
Numbers	3.1415926	<number>
Emoticon	:)	<smiley_face>
Username	@POTUS	<username>
Long words	greeeeeeat	great <elong>
Retweet Tag	RT:	Removed
Capitalization	SAD	<allcaps> sad

weights trained on another dataset to initialize our model can greatly reduce the risk of overfitting. We then fine-tune the Inception-Resnet model on Twitter images. In our model, we use the feature map before the final softmax layer as the image representation $f_{CNN}(I)$.

Textual LSTM-RNN The character limitation and the lightweight retweet operation noticeably differentiate the Twitter language to be very different from daily languages. Users tend to use abbreviations, Internet slang, emojis, and hashtags to emphasize their emotions (see Table 1 for an example). Thus, we need a powerful language model to characterize and understand the semantics of the tweeted text. Long-short term memory based recurrent neural networks (LSTM-RNNs) have recently demonstrated major success in different natural language processing tasks [35]. As a form of memory-based recurrent networks, it is natural for an LSTM-RNN to model variable length sequences such as tweet text. Individual words are first mapped to an embedding space by a word embedding layer. The sequence of embedding vectors are then fed through LSTM to extract textual features. We randomly initialize the word embedding layer and train it from scratch using only Twitter data to better model the Twitter specific language. We take the final output from the LSTM-RNN as the textual feature $f_{LSTM}(L)$ for tweet T.

Joint Embedding Language L and image I within a given tweet T are often related to each other. Standard CNN and LSTM models can extract rich feature representations from the photo I and language L separately, but they are not designed to discover and utilize the underlying cross-semantic relationships. Hence, if we just concatenate the extracted image and text feature vectors (similar to [19, 40]), they will belong to different embedding spaces and hence will not be very effective for popularity prediction. Therefore, we propose a nonlinear joint embedding network (see Figure 3) that maps the image feature $f_{CNN}(I)$ and the textual feature $f_{LSTM}(L)$ into a shared latent feature space where the two different data modalities are well correlated.

As shown in Figure 3, our proposed nonlinear joint embedding network consists of two branches. Each branch processes the input feature vector sequentially using a fully-connected layer, a Rectified Linear Unit (ReLU) activation layer, a second fully-connected layer, a batch normalization layer, and an L_2 normalization layer. The two branches are independently initialized and trained. For a tweet $T(I, L, U)$ with visual features $f_{CNN}(I)$ and language features $f_{LSTM}(L)$, the joint embedding network $g(f)$ maps them to a common latent feature space as $h(L) = g(f_{LSTM}(L))$ and $h(I) = g(f_{CNN}(I))$. $h(L)$ and $h(I)$ are L_2 normalized and are of the same dimensions. The two feature vectors $h(L)$ and $h(I)$ are concatenated and fed through two additional fully-connected layers to produce the joint content feature representation $F_d(L, I)$.

$$F_d(L, I) = \sigma(\mathbf{W}_2 \cdot (\sigma(\mathbf{W}_1 \cdot [h(L); h(I)] + \mathbf{b}_1)) + \mathbf{b}_2) \tag{4}$$

Given a training tweet $T_i(I_i, L_i, U_i)$, the joint embedding network will map the visual and textual features in a shared latent space ($H(I_i)$ and $H(L_i)$, respectively). Since the output of the embedding network is L_2 normalized, the Euclidean distance $d(I_i, L_i)$ is used to measure the similarity between image I_i and sentence L_i in the latent space. To discover and utilize the semantic relationships between the language domain and the image domain, we enforce a bi-directional distance constraint on the joint space. Similar to [39], we want the distance between an image I_i and its associated text L_i to be smaller than the distance between the image I_i and non-related text L_j by some enforced margin m:

$$d(I_i, L_i) + m < d(I_i, L_j), \qquad \forall j \neq i \tag{5}$$

Similarly, we would like to enforce that the distance between a sentence $L_{j'}$ and its associated image $I_{j'}$ is less than the distance between the sentence $L_{j'}$ and a non-related image $I_{k'}$ by the same margin m:

$$d(I_{j'}, L_{j'}) + m < d(I_{k'}, L_{j'}), \qquad \forall k' \neq j' \tag{6}$$

We combine the bidirectional constraints into a loss

function using the hinge loss:

$$L_{JE} = \frac{1}{M}\sum_{i,j,k}\{\max[0, m + d(I_i, L_i) - d(I_i, L_j)] + \alpha \max[0, m + d(I_i, L_i) - d(I_k, L_i)]\} \quad (7)$$

where m is a predefined margin, α is a predefined weighting scalar, and M is the total number of triplets. We set $m = 0.05$ and $\alpha = 1$ for all our experiments.

3.2. Social Features

Tweets are spread over Twitter by its users' retweet operations. The content quality of a tweet and the characteristics of its author can significantly affect its potential reach. Influential and active users can spread the word much faster and broader on the network than less well-connected users. Thus, it's natural to consider the authors' characteristics and potential influences on the network when predicting the popularity of a new tweet. We can directly extract social features from the author's profile U: `account_age`, `friend_count`, `follower_count`, `total_tweet_count`, `favorited_tweet_count`. Together with the cross-product transformation features $\phi(U) = \{u_i \cdot u_j | u_i \in U, u_j \in U, i < j\}$, we have the following social feature:

$$F_s(U) = [U; \phi(U)] \quad (8)$$

Compared with the textual and the visual features, the $F_c(U)$ features are of much lower dimensions and are much easier to interpret. The social features are used together with the content features to predict the retweet count.

3.3. Dynamics RNN

Temporal diffusion information are widely used for popularity prediction. Instead of using a reinforced Poisson process [30] or Hawkes Process [21], we employ a simple but effective recurrent neural network to learn the temporal propagation pattern. Compared with other diffusion based models [21, 30], our dynamics RNN can easily integrate content and social features.

Given a tweet $T(I, L, U, D)$, due to data collection limitations, the propagation data D is not uniformly sampled in the temporal domain. We first use linear interpolation to uniformly resample the propagation process D in the temporal domain using a fixed time interval. At each time step i, the dynamics RNN updates its hidden state h_i and computes an output prediction $\tilde{\lambda_i}$ by iterating the following relations:

$$\begin{aligned} c &= W_{hc}[F_c(L, I), F_s(U)] \\ h_i &= tanh(W_{hr} r_{i-1} + W_{hh} h_{t-1} + b_h + c \odot \mathbb{I}[i = 0]) \\ \ln(\lambda_i) &= W_{oh} h_i + b_o \end{aligned} \quad (9)$$

Table 2: Detailed configurations of our proposed network.

Component	Layer	Dimension/Units
CNN	Output	1792
LSTM	WE	512
	Hidden	512
Joint Embedding	FC(W1)	768
	FC(W2)	256
	FC(V1)	512
	FC(V2)	256
	Concat	512
	FC	256
	FC	128
Social Features	-	25
Dynamics RNN	Hidden	256

Weights $W_{hc}, W_{hr}, W_{hh}, W_{oh}$ and biases b_h, b_o are learnable parameters. $\mathbb{I}$ is an indicator function. We found that conditioning the dynamics RNN at its first step works better than conditioning it at every time step i.

3.4. Poisson Regression

We train our model to maximize the Poisson likelihood given a collection of N training tuples of tweets T_i and their retweet counts $r_{gt,i}$:

$$\theta^* = \arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N}\{r_{gt,i} \ln \lambda(T_i) + \lambda(T_i)\} \quad (10)$$

where θ contains all parameters of our proposed model. The loss function can be denoted as:

$$L_{Poisson} = \frac{1}{N}\sum_{i=1}^{N}\{r_{gt,i} \ln \lambda(T_i) + \lambda(T_i)\} \quad (11)$$

4. Experiments

We study the contributions of each data modality, and each network component on multiple datasets. We compare our model against several state-of-the-art methods. Please refer to the supplementary materials for more details on preprocessing, filtering, visualization, etc.

4.1. Network Architecture and Training

Network configuration We detail the architecture and configuration of our proposed model in Table 2. Our overall network contains multiple components. It's challenging to train the entire model from scratch in an end-to-end fashion. Therefore, we first train each individual component separately. After each component has reached a stable state, the entire model can be trained jointly.

Warm-up language model We start by training the twitter language model, i.e, the word embedding layer and the LSTM network. Both word embedding and LSTM are randomly initialized. A generalized linear model is used to predict the Poisson parameter λ from the LSTM output vector.

$$\ln(\lambda(T)) = \mathbf{w} \cdot f_{LSTM}(L) + b \quad (12)$$

where linear weight $\mathbf{w}$ and bias b are learnable parameters. We then train the LSTM network to directly predict the hidden variable λ based on Equation (12) with the negative log-likelihood loss function in Equation (11). The gradient magnitude is clipped to 5 during back-propagation to avoid the gradient explosion problem. We train the LSTM with Poisson loss for 100k iterations.

Fine-tune the CNN We then fine-tune the CNN weights on Twitter images. Similar to Equation (12), we use a generalized linear model to predict the hidden Poisson parameter from the CNN feature output $f_{CNN}(I)$. The Poisson loss (Equation (11)) is used as the objective for fine-tuning.

Learn joint embedding We use the pre-trained CNN feature concatenated with the warmed-up LSTM output as the input to train the joint embedding network. The joint embedding network is also randomly initialized. During this initial training phase, both the CNN and the LSTM network are fixed. Only the joint embedding network is trainable. Unlike [39], we only adopt the bi-directional constraints and relax the structural constraints, since only one sentence/image pair exists within each tweet.

We randomly initialize all the layers of the joint embedding network and optimize them wrt L_{JE}. Triplets containing an associated image/sentence pair (I_i, L_i), a non-relating image I_j, and a non-relating sentence L_k are used to optimize the loss function L_{JE}. However, it is computationally prohibitive to optimize the loss function L_{JE} over all possible triplets in the dataset. Thus we follow a similar approach to [39] to sample triplets within each mini-batch of the training dataset during optimization. We first compute the similarity distance $d(I_i, L_i)$ for all tweets within the batch. For each tweet (a ground-truth image/text pair), we then find the top K non-relating images and the top K non-relating sentences violating the bi-directional constraint. We use $K = 20$ in all the experiments. We then train the joint embedding network with the CNN and LSTM being fixed.

Warm-up dynamics RNN We randomly initialize the dynamics RNN. The dynamics RNN is designed to predict the hidden Poisson parameter λ from past observations. Thus we train the dynamics RNN using the Poisson loss Equation (11). We fix the CNN, language LSTM, and the joint embedding network during the warm-up phase of the dynamics RNN and train it for for 100k iterations. The gradient magnitude is clipped to 5 during training.

Preventing overfitting Real-world Twitter data can be very noisy. We adopted multiple techniques, to avoid overfitting the noisy training data. Similar to [38], initializing the CNN using pre-trained weights greatly helps to prevent overfitting. We also use dropout layers in the LSTM network. Each fully-connected layer in the joint embedding model is also followed by a dropout layer. The keep probability of all dropout layers is 0.7. Additionally, L_2 regularization is applied during training.

Optimization We initialize and warm up each component of our network separately as discussed above. Then we combine the negative log-likelihood $L_{Poisson}$ (Equation (11)), the joint embedding loss L_{JE} (Equation (7)), and the weight θ regularization as the joint loss function:

$$L = L_{Poisson} + \kappa_1 L_{JE} + \kappa_2 \|\theta\|_2 \tag{13}$$

Weight parameters $\kappa_1 = 0.5, \kappa_2 = 0.05$ are selected via cross-validation. We train the network end-to-end by minimizing the above loss function (Equation (13)).

Our model,implemented in TensorFlow, is optimized using Adam [20] on three nVidia K20 GPUs. We use a learning rate of 10^{-5}. The learning rate decays every $100k$ iterations with an exponential rate of 0.9.

4.2. Datasets

We train our model and evaluate their prediction accuracy on multiple Twitter datasets collected from real-world Twitter streams across different time periods.

Our model employed an LSTM network to learn an underlying language model for tweets. In principle, it would require a dedicated LSTM network for each language used on Twitter. Without loss of generality, we only studied the popularity prediction problem for English tweets.

MBI-1M The MicroBlog-Images (MBI-1M) dataset [6] collected in 2013 contains 1 million tweets. Retweet counts and favorite counts for the contained tweets were collected later in 2014. Only English tweets containing images from the MBI-1M dataset are used in our experiments. We follow [6] to split the English tweets from the MBI-1M dataset into 70% training, 10% validation, and 20% test sets respectively.

Twitter2015 We also collected over 40 million tweets from Nov. 2015 to Apr. 2016 using the Twitter API. The Twitter streaming API returns a small random set of all public tweets (up to 1%). Similarly, we only used tweets that are written in English and contain at least one image. We randomly split the Twitter2015 dataset into 80% training, 10% validation, and 10% testing sets.

Twitter2016 Topics on Twitter change rapidly and continuously over time. Machine learning based approaches must have good generalization capabilities to deal with such rapid topic drift. To evaluate the generalization capabilities, we collected another dataset using the Twitter public API in Oct. 2016. This dataset contains 9 million English tweets. We reserve this entire dataset for testing purposes only. The maximum retweet count encountered when the data collection ends is used as the ground-truth retweet value for Twitter2015 and Twitter2016 dataset. Detailed statistics for the three datasets can be found in the supplementary materials.

TemporalTwitter2015 Due to the limited sampling ratio of the Twitter public API. we can only collect partial

propagation data. During the Twitter2015 collection period, we recorded tweets with over 50 retweeted sampling points and assemble them into a new TemporalTwitter2015 dataset. Tweets propagating longer than 72 hours are discarded. The TemporalTwitter2015 contains 12,187 valid tweets.

4.3. Dataset Preprocessing

We resize the images to 299×299 pixels to be compatible with the Inception-Resnet model.

Messages on Twitter usually contain informal language. Accordingly, we preprocess the text to reduce irregularities to lessen the burden on the later LSTM network. We first reduce the irrelevant information in tweet text by simplifying hashtags, numbers, usernames, etc. Please refer to Table 1 for detailed pre-processing rules. URLs embedded in tweets usually point to external resources. According to [4], URLs elicited more positive feelings or rated more interesting were more likely to spread. Hence instead of using a single symbolic word `<URL>`, we expanded and parsed the hashed/shortened URL within tweets. Only domain names are recorded as words. After the text is pre-processed, we tokenize the pre-processed text string into words and build a Twitter vocabulary. Rare words appearing no more than 10 times in the corpus are replaced by a symbolic word `<unknown>` in the vocabulary. Our vocabulary contains over 500k distinct words.

4.4. Popularity Prediction Evaluation

Evaluation metric We evaluate our proposed method on the aforementioned datasets and compare our results against multiple state-of-the-art methods [6, 24, 25, 19]. The Spearman's ranking correlation and mean absolute percentage error (MAPE) are adopted as the evaluation metric.

Static Setting Evaluation We first evaluate our multi-modal regression method in the static setting. We compare our multi-modal model with state-of-the-art content-based methods [6, 24, 25, 19]. For fair comparison, we discard the dynamics RNN and only use Poisson regression on the joint content and social features. More formally $\ln(\lambda) = W[F_c(L, I), F_s(U)] + b$ is used for prediction.

Table 4 demonstrates that our proposed joint model has superior performance compared to other content-based methods. Compared with our model, [25] only use simple visual features such as scene categories, the number of human faces, and color information. [6] and [19] were originally proposed to predict online photo popularities. Neglecting textual information hinders its performance in tweet popularity prediction tasks. [24] utilized visual, textual, and social cues to predict brand-related popularities, thus outperforming the other three baseline methods. Compared to the baseline content-based methods, our model not only utilizes more advanced feature representations, but also a joint embedding model to maximize the correlation across modalities, which helped us outperform the state-of-the-art.

Dynamic Setting Evaluation We evaluate our dynamic-RNN model on the TemporalTwitter2015 datasets against the state-of-the-art TiDeH method [21]. For a tweet, the retweet count at 72 hours after its issue is predicted. See Table 6 for quantitative results.

Using propagation data alone, the simple RNN model demonstrated slightly inferior performance compared to the baseline method. However, by properly combining content features and social cues, our model can achieve slightly better prediction accuracy than baseline methods. Utilizing all available data modalities (image I, text L, social cue S, and propagation information D), as well as the proper Poisson loss, contributed to the performance improvement.

4.5. Ablation Studies

We thoroughly studied the prediction performance with different loss functions and different joint modeling methods. Detailed statistics can be found in Table 5 and Table 6.

Compared with simple linear loss, Poisson loss can improve prediction performance on different data modalities. Poisson distribution is more suitable to model discrete data distributions, thus outperforming the simple linear loss.

Social features generally outperform visual and textural features when used in isolation. Our observation agrees with the literature [40]. However, naively concatenating features from different modalities does not significantly improve the performance over simple social features or dynamics features. However, our Poisson regression model and our joint embedding are key to our performance improvement. By explicitly aligning different modality features in the common space, the regression model can pay more attention to the common salient features, and neglects the differences between the image and text description. Table 5 shows that combining textual or visual features with social cues can outperform all single modality. Thus, both visual and textual features can benefit social cues when predicting the popularities.

On the TemporalTwitter2015 dataset, the dynamics feature when used alone, outperforms both visual and textual features. When properly combined with content based features, we achieve the best performance on the evaluation dataset. Diffusion based methods require a sequence of early retweeting/propagation observations to predict future message outreach. Compared with content based methods, such early observations are hard or sensitive to acquire, limiting the practical applicability of diffusion based methods. Being able to make the prediction based on content alone, or combining content into the diffusion models, is of great practical importance.

Table 3: Dataset statistics. For all the three datasets, we first filter for English tweets (the English column). Then we discard tweets without visual images (English+Image column). If we capture multiple retweets of the same tweet, we group them as one tweet and record its maximum retweet number (the Unique Tweets column). Such filters help us remove redundancies in the datasets and make training time manageable.

Dataset	Collection Time	Total	English	English + Image	Unique Tweets	Unique Users
MBI1M [6]	2013	1,007,197	347,865	347,865	347,865	318,591
Twitter2015	2015	40,467,493	13,651,796	3,104,566	1,886,498	475,291
Twitter2016	2016	32,173,022	9,655,915	1,923,507	1,076,958	350,519

Table 4: Comparison against state-of-the-art baseline methods. By using advanced CNN and LSTM models and joint embedding, our method outperform previous approaches. Spearman: higher is better. MAPE: lower is better.

Method	Spearman			MAPE		
	MBI1M	T2015	T2016	MBI1M	T2015	T2016
McParlane et al	0.188	0.269	0.257	0.093	0.121	0.137
Khosla et al	0.185	0.273	0.254	0.097	0.103	0.124
Cappallo et al	0.189	0.265	0.258	0.089	0.095	0.119
Mazloom et al	0.190	0.287	0.262	0.073	0.097	0.117
Ours	**0.229**	**0.358**	**0.350**	**0.057**	**0.084**	**0.103**

Table 5: Quantitative evaluation of each data modality. 'V': visual, 'T': textual, 'S': social features. 'L' = linear loss, 'P' = Poisson loss. 'FC' = fully-connected layers without joint embedding, 'Joint' = joint embedding model. For multi-modal FC models, features from different modalities are concatenated together. Spearman: higher is better. MAPE: lower is better.

Feature	Model	Loss	Spearman			MAPE		
			MBI1M	T2015	T2016	MBI1M	T2015	T2016
V	FC	L	0.149	0.248	0.232	0.147	0.152	0.157
T	FC	L	0.157	0.267	0.248	0.132	0.140	0.145
S	FC	L	0.175	0.281	0.269	0.113	0.128	0.130
V	FC	P	0.163	0.278	0.261	0.135	0.149	0.153
T	FC	P	0.172	0.283	0.275	0.129	0.138	0.142
S	FC	P	0.181	0.301	0.289	0.103	0.125	0.129
TS	FC	P	0.198	0.325	0.319	0.090	0.109	0.116
VS	FC	P	0.193	0.321	0.313	0.092	0.111	0.118
VTS	FC	L	0.188	0.311	0.294	0.097	0.112	0.119
VTS	FC	P	0.212	0.341	0.327	0.083	0.103	0.115
VTS	Joint	L	0.207	0.339	0.325	0.071	0.097	0.112
VTS	Joint	P	**0.229**	**0.358**	**0.350**	**0.057**	**0.084**	**0.103**

4.6. Attributes Analysis

To gain more insights on the influencing factors leading to the popularity of tweets, we analyze the common attributes of highly retweeted posts. We first manually labeled images and sentences with an attribute set. The common attributes of the highly scoring tweets are then analyzed.

For visual features, the following attributes are manually collected on 5K images: *dynamic GIF, animal, human, beautiful, not beautiful, sexual, containing text, synthetically generated*. We notice the following attributes to be highly correlated with the virality of tweets: *animal, not beautiful, sexual, containing text, synthetically generated*. Especially, images containing text are quite popular. Users also like to generate images by composing multiple images, or adding textual descriptions in the image. Such synthetic generated or augmented images are likely to go viral.

Table 6: Quantitative evaluation of dynamic propagation features. 'V': visual, 'T': textual, 'S': social features, 'D': dynamic features. 'L' = linear loss, 'P' = Poisson loss. 'FC' = fully-connected layers without joint embedding, 'Joint' = joint embedding model. Spearman: higher is better. MAPE: lower is better.

Feature	Loss	Model	Spearman	MAPE
V	L	FC	0.217	0.152
T	L	FC	0.223	0.147
S	L	FC	0.247	0.139
D	L	FC	0.290	0.109
V	P	FC	0.232	0.142
T	P	FC	0.241	0.129
S	P	FC	0.260	0.120
D	P	FC	0.297	0.097
TD	P	FC	0.317	0.096
VD	P	FC	0.320	0.097
SD	P	FC	0.339	0.095
VTSD	L	FC	0.310	0.095
VTSD	P	FC	0.349	0.091
VTSD	L	Joint	0.357	0.089
VTSD	P	Joint	**0.366**	**0.085**
TiDeH	-	-	0.364	0.087

For textual attributes, we labeled 5K sentences with the following attributes: *political, religious, emotional, having emoji, having Twitter slang, having URL*. We found *political* and *URL* to be influential. Emoji expressions and slangs are "ubiquitous" on Twitter, thus not providing extra information for popularity prediction. URLs may contain extra information that leads to users' retweet actions.

5. Conclusion

In this paper, we studied the problem of predicting tweet popularity. Our method estimates the potential reach of a tweet based on its image, language, author relationships, and propagating behaviors. We show that naively combining multimodal features does not improve upon social features but via a joint embedding model, our Poisson regression approach not only shows complementary improvements, but also achieves state-of-the-art results on multiple datasets. We evaluated our model on Twitter data but our proposed method is also applicable to other social networks.

Acknowledgements: Supported in part by the NSF No. IIS-1349074, No. CNS-1405847.

References

[1] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini. A peek into the future: Predicting the evolution of popularity in user generated content. In *WSDM*, 2013. 2

[2] G. Andrew, R. Arora, J. Bilmes, and K. Livescu. Deep canonical correlation analysis. In *ICML*, 2013. 3

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 3

[4] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts. Everyone's an influencer: Quantifying influence on twitter. In *WSDM*, 2011. 7

[5] H. Becker, M. Naaman, and L. Gravano. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 2011. 2

[6] S. Cappallo, T. Mensink, and C. G. Snoek. Latent factors of visual popularity prediction. In *International Conference on Multimedia Retrieval*, 2015. 2, 6, 7, 8

[7] J. Cheng, L. Adamic, P. A. Dow, J. M. Kleinberg, and J. Leskovec. Can cascades be predicted? In *WWW*, 2014. 2

[8] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv 1412.3555*, 2014. 3

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009. 3

[10] L. Derczynski, A. Ritter, S. Clark, and K. Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, 2013. 2

[11] A. Deza and D. Parikh. Understanding image virality. In *CVPR*, 2015. 2

[12] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. A. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 3

[13] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are you talking to a machine? dataset and methods for multilingual image question answering. In *NIPS*, 2015. 3

[14] A. Go, R. Bhayani, and L. Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[16] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 2

[17] K. Ishiguro, A. Kimura, and K. Takeuchi. Towards automatic image understanding and mining via social curation. In *ICDM*, 2012. 2

[18] A. Java, X. Song, T. Finin, and B. Tseng. Why we twitter: Understanding microblogging usage and communities. In *Proceedings of the 9th WebKDD and 1st SNA-KDD 2007 Workshop on Web Mining and Social Network Analysis*, 2007. 2

[19] A. Khosla, A. Das Sarma, and R. Hamid. What makes an image popular? In *WWW*, 2014. 2, 4, 7

[20] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 6

[21] R. Kobayashi and R. Lambiotte. Tideh: Time-dependent hawkes process for predicting retweet dynamics. *arXiv:1603.09449*, 2016. 5, 7

[22] H. Kwak, C. Lee, H. Park, and S. Moon. What is twitter, a social network or a news media? In *WWW*, 2010. 2

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[24] M. Mazloom, R. Rietveld, S. Rudinac, M. Worring, and W. van Dolen. Multimodal popularity prediction of brand-related social media posts. In *ACM MM*, 2016. 7

[25] P. J. McParlane, Y. Moshfeghi, and J. M. Jose. "nobody comes here anymore, it's too crowded"; predicting image popularity on flickr. In *ICMR*, 2014. 2, 7

[26] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv 1301.3781*, 2013. 3

[27] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur. Recurrent neural network based language model. *Interspeech*, 2010. 3

[28] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011. 3

[29] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *IEEE TPAMI*, 2016. 2

[30] H.-W. Shen, D. Wang, C. Song, and A.-L. Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. *arXiv:1401.0778*, 2014. 5

[31] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2013. 3

[32] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, 2011. 3

[33] N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*. Curran Associates, Inc., 2012. 3

[34] B. Suh, L. Hong, P. Pirolli, and E. H. Chi. Want to be retweeted? large scale analytics on factors impacting retweet in twitter network. In *2010 IEEE Second International Conference on Social Computing (SocialCom)*, 2010. 1

[35] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 3, 4

[36] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv:1602.07261*, 2016. 2, 3

[37] C. Tan, L. Lee, and B. Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *ACL*, 2014. 2

[38] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE TPAMI*, 2016. 3, 6

[39] L. Wang, Y. Li, and S. Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 3, 4, 6

[40] K. Yamaguchi, T. L. Berg, and L. E. Ortiz. Chic or social: Visual popularity analysis in online fashion networks. In *ACM MM*, 2014. 1, 2, 4, 7

[41] Q. Zhao, M. A. Erdogdu, H. Y. He, A. Rajaraman, and J. Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *KDD*, 2015. 1, 2

[42] C. Zhong, D. Karamshuk, and N. Sastry. Predicting pinterest: Automating a distributed human computation. In *WWW*, 2015. 2