Retweet Wars: Tweet Popularity Prediction via Multimodal Regression

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

Abstract

In this paper, we re-visit the tweet popularity prediction problem by considering all available data modalities: tweet text semantics, embedded visual content, and authors' social influence. To better model the tweet content, a joint-embedding network is proposed to combine different cues together. The network is optimized by a novel Poisson regression loss. We demonstrate that content based features can be used to improve upon social 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.

Introduction

The world is better connected than ever before. 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 content 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.

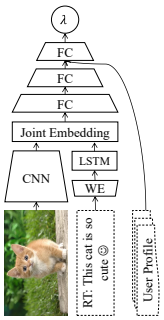
Popularity prediction has long interested various research communities [30, 4, 5, 13, 16, 19, 8]. Social features and cascading process modeling are the dominant foundations for previous works [1, 6, 15, 37, 38]. Recently, deep learning based methods have revolutionized many vision and language tasks [14, 20, 23, 24, 26, 7, 27, 31]. Hence, the question arises whether such deep visual and textual representations can help improve upon the popularity prediction accuracy? In this paper, we study the role of content for the popularity prediction task. 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, 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.

To summarize, our main contributions are: 1) A tri-modal neural network model that harnesses all available Twitter data modalities: visual, textual, and social 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) Ablation and attribute analysis to explain model component and modality contributions.

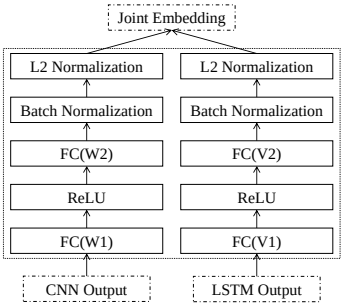
Methodology

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 text L is first issued by its author U. The maximum retweet count during the data collection period is used as the ground-truth retweet count r_{gt}.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.



(a) Overall architecture.



(b) Joint embedding network.

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

(c) Detailed network architecture.

Figure 1: Network architecture.

Multimodal Network We adopted the Inception-Resnet architecture [32] to extract feature representations from the Twitter images I. Using weights trained on large-scale dataset to initialize our model can greatly reduce the risk of overfitting. We then fine-tune the Inception-Resnet model on Twitter images. We use the feature map before the final softmax layer as the image representation $f_{CNN}(I)$.

Twitter language is very different from daily languages. Thus, we used a powerful Long-short term memory (LSTM) network to model the Twitter language. 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.

Language L and image I within a given tweet T are often related to each other. Thus, we propose a joint embedding network to map the image feature $f_{CNN}(I)$ and the textual feature $f_{LSTM}(L)$ into a common semantic space where the two different data modalities are better correlated. Thus, we enforce a bi-directional distance constraint on the joint space. Similar to [35], 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: $d(I_i, L_i) + m < d(I_i, L_j), \forall j \neq i$. 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: $d(I_{j'}, L_{j'}) + m < d(I_{k'}, L_{j'}), \forall k^{'} \neq j^{'}$. We combine the bidirectional constraints into a loss function using the hinge loss:

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

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.

Social Features Tweets are spread over Twitter by users' retweet operations. Influential users can spread the word much faster and broader than less well-connected users. Thus, we consider to use 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: $F_s(U) = [U; \phi(U)]$. Compared with the textual and the visual features, the social 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.

Poisson Regression The retweet count r of a tweet $T(I, L, U)$ follows a Poisson distribution $P(R = r|\lambda) = e^{-\lambda}\lambda^{-r}/r!$, where the latent variable $\lambda \in \mathbb{R}^+$ defines the mean and variance of the underlying Poisson distribution. Our proposed network combines multi-modal information from the unseen tweet $\tilde{T}$ to model 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 $P(R; \tilde{\lambda})$: $\tilde{r} = \max(\lceil \tilde{\lambda} \rceil - 1, \lfloor \tilde{\lambda} \rfloor)$. Thus, we train our model to maximize the Poisson likelihood given a collection of N training tuples of tweets T_i and their retweet counts $r_{gt,i}$: $L_{Poisson} = \frac{1}{N} \sum_{i=1}^{N} \{r_{gt,i} \ln \lambda(T_i) + \lambda(T_i)\}$.

Table 1: 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. [22]	0.188	0.269	0.257	0.093	0.121	0.137
Khosla et al. [17]	0.185	0.273	0.254	0.097	0.103	0.124
Cappallo et al. [5]	0.189	0.265	0.258	0.089	0.095	0.119
Mazloom et al. [21]	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**

Experiments

Training Our overall network contains multiple components. It's challenging to train the entire model from scratch in an end-to-end fashion. Thus, we first train each individual component separately. After each component has reached a stable state, the entire model can be trained jointly.

We start by training the twitter language model, i.e, the word embedding and the LSTM network. We optimize the LSTM network to directly predict the Poisson parameter λ of the Poisson loss function. The gradient magnitude is clipped to 5 during back-propagation to avoid gradient explosion. We train the LSTM with Poisson loss for 100k iterations.

We then fine-tune the CNN weights on Twitter images for 100k iterations. Similar to LSTM warm-up, we use a generalized linear model to predict the hidden Poisson parameter from the CNN feature output $f_{CNN}(I)$. The Poisson loss is used as the objective for fine-tuning.

We use the fine-tuned CNN feature together with the warmed-up LSTM output as the input to train the joint embedding network. We then train the joint embedding network with the CNN and LSTM being fixed. Similar to Wang et al. [35], triplets are sampled within each mini-batch of the training dataset during optimization. We first compute the Euclidean 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.

Real-world Twitter data can be very noisy. We adopted multiple techniques, to avoid overfitting the noisy training data. Similar to Vinyals et al. [34], 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.

We initialize and warm up each component of our network separately as discussed above. Then we combine the negative log-likelihood $L_{Poisson}$, the joint embedding loss L_{JE} (Equation (1)), and the weight θ regularization as the joint loss function $L_{joint} = L_{Poisson} + \kappa_1 L_{JE} + \kappa_2 \|\theta\|_2$. 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 L_{joint}. Our model,implemented in TensorFlow, is optimized using Adam [18] 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.

Datasets We train our model and evaluate their prediction accuracy on multiple Twitter datasets collected from real-world Twitter streams across different time periods. Without loss of generality, we only studied the popularity prediction problem for English tweets.

The MicroBlog-Images (MBI-1M) dataset [5] collected in 2013 contains 1 million tweets. We also collected two datasets from Twitter in 2015 and 2016 respectively. We randomly split the Twitter2015 dataset into 80% training, 10% validation, and 10% testing sets. The entire Twitter2016 dataset is reserved for testing of the model generalization capability. Please refer to the supplementary materials for detailed dataset statistics and visualizations.

Messages on Twitter usually contain informal language. Accordingly, we preprocess the text to reduce irregularities. We first reduce the irrelevant information in tweet text by simplifying hashtags, numbers, usernames, etc.We expanded and parsed the hashed/shortened URL within tweets. Only domain names are recorded as words. Then we tokenize the pre-processed text into words and build a Twitter vocabulary. Rare words appearing no more than 10 times in the corpus are discarded. Our vocabulary contains over 500k distinct words.

Table 2: 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**

Evaluation We evaluate our proposed method on the aforementioned datasets and compare our results against multiple state-of-the-art methods [5, 21, 22, 17]. The Spearman's ranking correlation and mean absolute percentage error (MAPE) are adopted as the evaluation metric. Table 1 demonstrates that our proposed joint model has superior performance compared to other content-based methods. Compared with our model, McParlane et al. [22] only use simple visual features such as scene categories, the number of human faces, and color information. Cappallo et al. [5] and Khosla et al. [17] were originally proposed to predict online photo popularities. Neglecting textual information hinders its performance in tweet popularity prediction tasks. Mazloom et al. [21] 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.

Analysis We thoroughly studied the prediction performance with different loss functions and different joint modeling methods. Detailed statistics can be found in Table 2. 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 [36]. Intuitively, textual description and visual image in a given tweet describe related content. But naively concatenating features from different modalities does not significantly improve the performance over simple social features or dynamics features. However, by jointly embedding them, the common semantic concepts can be emphasized, which can help the network to focus on semantically important "concepts/content" for popularity prediction. Similar intuition is also reported by VQA methods [11]. Thus, our Poisson regression model and our joint embedding are key to our performance improvement. Table 2 shows that combining textual or visual features with social cues can outperform all single modality. Both visual and textual features can benefit social cues when predicting the popularities.

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, and author relationships. By using a joint embedding model together with the novel Poisson regression loss, our method demonstrated complementary improvements and state-of-the-art results compared to naive feature concatenation.

In the supplementary material, we present more analysis, visualization, and discussion about the Twitter dataset and popularity prediction task: 1) we reviewed relevant works on popularity predictions, multi-modal deep learning, and other related fields; 2) we analyzed different attributes of tweets to provide insight for the popularity prediction task; 3) we further extend our model to take propagation data as input. An RNN module is used to model the dynamic tweet propagation process.4) we provide further details, analyses, and visualizations of our utilized Twitter datasets; 5) we list both positive and negative tweet examples with our predictions.

References

[1] Mohamed Ahmed, Stella Spagna, Felipe Huici, and Saverio Niccolini. A peek into the future: Predicting the evolution of popularity in user generated content. In *WSDM*, 2013.

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, 2013.

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

[4] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 2011.

[5] Spencer Cappallo, Thomas Mensink, and Cees G.M. Snoek. Latent factors of visual popularity prediction. In *International Conference on Multimedia Retrieval*, 2015.

[6] Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *WWW*, 2014.

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv 1412.3555*, 2014.

[8] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, 2013.

[9] Arturo Deza and Devi Parikh. Understanding image virality. In *CVPR*, 2015.

[10] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc\textquotesingle Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013.

[11] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Conference on Empirical Methods in Natural Language Processing*, pages 457–468. ACL, 2016.

[12] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? dataset and methods for multilingual image question answering. In *NIPS*, 2015.

[13] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

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

[16] Akshay Java, Xiaodan Song, Tim Finin, and Belle 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.

[17] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *WWW*, 2014.

[18] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014.

[19] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *WWW*, 2010.

[20] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016.

[21] Masoud Mazloom, Robert Rietveld, Stevan Rudinac, Marcel Worring, and Willemijn van Dolen. Multimodal popularity prediction of brand-related social media posts. In *ACM MM*, 2016.

[22] Philip J. McParlane, Yashar Moshfeghi, and Joemon M. Jose. "nobody comes here anymore, it's too crowded"; predicting image popularity on flickr. In *ICMR*, 2014.

[23] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. *Interspeech*, 2010.

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv 1301.3781*, 2013.

[25] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *ICML*, 2011.

[26] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE TPAMI*, 2016.

[27] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, 2011.

[28] Richard Socher, Andrej Karpathy, Quoc V. Le, Chris D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2013.

[29] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*. Curran Associates, Inc., 2012.

[30] 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.

[31] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014.

[32] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv:1602.07261*, 2016.

[33] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *ACL*, 2014.

[34] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE TPAMI*, 2016.

[35] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016.

[36] Kota Yamaguchi, Tamara L Berg, and Luis E Ortiz. Chic or social: Visual popularity analysis in online fashion networks. In *ACM MM*, 2014.

[37] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *KDD*, 2015.

[38] Changtao Zhong, Dmytro Karamshuk, and Nishanth Sastry. Predicting pinterest: Automating a distributed human computation. In *WWW*, 2015.

Retweet Wars: Tweet Popularity Prediction via Multimodal Regression Supplementary Materials

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

Contents

In this supplementary material, we present more analysis, visualization, and discussion about the Twitter dataset and popularity prediction task.

1. We first review relavent works on popularity predictions, multi-modal deep learning, and other related fields in Section 1.
2. We analyze different attributes of tweets to provide insight for the popularity prediction task in Section 2.
3. We further extend our model to take propagation data as input in Section **??**. We propose to use a recurrent neural network (RNN) to model the dynamic tweet propagation process. The content based feature is used as pre-conditioning of the dynamic RNN.
4. We provide further details, analyses, and visualizations of our utilized Twitter datasets in Section 4.

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

5. We list several positive and negative tweet examples with our retweet predictions in Section 5.2.

1 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 [15] and [18], 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 [4], sentiment classification [12], popularity prediction [5], and tweet-based language processing [8]. 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, [21] utilized textual, visual, and social cues to predict the image popularity on Flickr. [16] used contextual and deeply-learned visual features to explore the factors influencing an online photo's popularity. [34] combined visual, textual, and social features to predict popularity in the fashion domain but only use tag-based text and no joint embedding models. [9] showed that mid-level image features trained on deep networks improved the performance of image virality prediction. [31] 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 [16, 34], 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. [14] showed social-oriented features were the best performer to predict image popularity on Twitter. [36] 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. [6] used temporal and structural features to predict cascades of photo shares on Facebook. [35] 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 [13], semantic segmentation [25], and object detection [19]. Deep-learning-based methods have also influenced natural language processing (NLP), from word embeddings [23] and language modeling with recurrent neural networks (RNNs) [22] to syntactic parsing [27] and machine translation [7, 30]. 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, [24] used deep autoencoder models to learn multi-modal features for audio-visual speech classification tasks. [29] 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 [28, 32] and visual question answering [3, 11]. [2] proposed a deep learning based extension to canonical correlation analysis (CCA). [10] used ranking losses to learn the linear transformations on visual and textual features. In our work, we follow the lead of [33] 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.

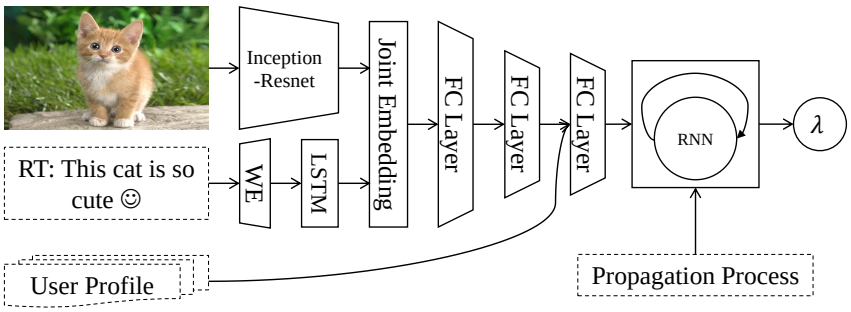


Figure 1: Our proposed multi-modal model to predict tweet popularity. A state-of-the-art Inception-Resnet CNN model is used to extract visual features and an LSTM is used to extract textural features. Visual and textual representations are then mapped to a common space by a joint embedding network. For 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 pre-condition the dynamics RNN, which predicts the Poisson model by looking at early stage propagation data.

2 Attribute 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.

For textual attributes, we labeled 5K sentences with the following attributes: *political, religious, emotional, having emoji, having Twitter slang, having URL*. We discovered that the following attributes to be highly correlated with tweet popularity: *political, containing URL*. 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.

3 Temporal Extension

Temporal diffusion information are widely used for popularity prediction. Instead of using a reinforced Poisson process [26] or Hawkes Process [17], we employ a simple but effective recurrent neural network to learn the temporal propagation pattern. Compared with other diffusion based models [17, 26], our dynamics RNN can easily integrate content and social features.

3.1 Dynamics RNN

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}.

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

Table 1: 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

following relations:

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

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. Notice that $\ln(\lambda)$ is used to ensure that $\lambda > 0$. 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.

3.2 Dynamic Scenario Evaluation

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:

$$L_{Poisson} = \frac{1}{N} \sum_{i=1}^{N} \{r_{gt,i} \ln \lambda(T_i) + \lambda(T_i)\} \tag{2}$$

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.

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.

Dynamic Setting Evaluation We evaluate our dynamic-RNN model on the TemporalTwitter2015 datasets against the state-of-the-art TiDeH method [17]. For a tweet, the retweet count at 72 hours after its issue is predicted. See Table 1 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.

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.

4 Dataset

4.1 Dataset Overview

We collected the Twitter2015 and Twitter2016 dataset using the Twitter public streaming API. We also used the MBI1M dataset. Detailed statistics can be found in Table 2.

Table 2: 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.

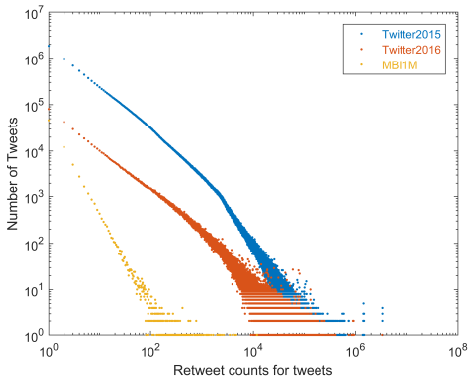
Dataset	Time	Total	English	English + Image	Unique Tweets	Unique Users
MBI1M [5]	2013	1,007,197	347,865	347,865	347,865	318,591
Twitter2015	2015	40,467,493	13,651,796	3,104,566	1,886,498	475,291
Twitter2016	2016	32,173,022	9,655,915	1,923,507	1,076,958	350,519

4.2 Visualizations

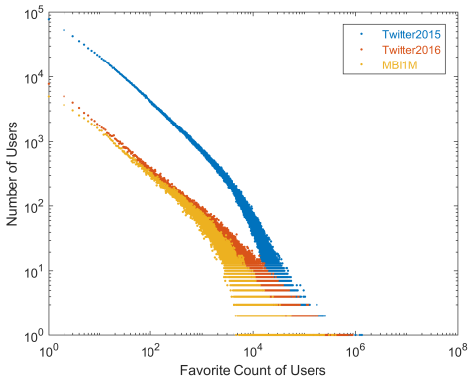
Notice in Figure 2c that the y-axis is the percentages of the users. Since the three datasets consist of different number of users, only a normalized histogram generates interpretable visualizations. Especially, the MBI1M dataset was collected in 2013, which explains that no user account younger than 3 years exists in MBI1M dataset. Twitter was founded in 2006, which agrees with our observation in Figure 2c that no user accounts older than 10 years exist. An interesting fact that we found in the user account history plot is that there is an obvious surge in user account numbers two years after Twitter went public. The actual reason that led to such user increase is beyond our knowledge.

Twitter is a social network, but the relationship between users on Twitter can be asymmetric. A user U_i can *follow* another user U_j, then U_i will receive all tweets/retweets from user U_j. U_j doesn't have to be friend with U_i to make U_j's messages spread to U_i. Two users can be *friends* if they follow each other. Such "one-way" friendship on Twitter is different from many other social network websites and thus brings Twitter more flexibility in disseminating information.

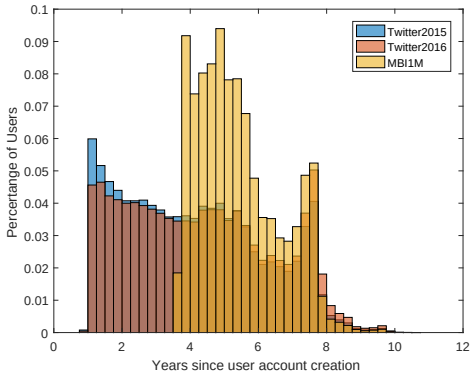
We visualize the user *follower* distribution in Figure 2f. Figure 2f clearly demonstrates the "power-law" distribution characteristics commonly seen on social networks [15]. Similarly, we visualize the user *friend* distribution in Figure 2d. Interestingly, all three datasets demonstrate multiple strikes in the plot. Actual Twitter following rules lead to such strikes. Twitter has explicit limits on the number of followers for each user. In October 2015, Twitter increased the follow limit from 2,000 to 5,000. Beyond 5,000 user accounts, there are limits to the number of additional users one can follow. Twitter chose to limit the following behavior to improve the site performance and reliability.



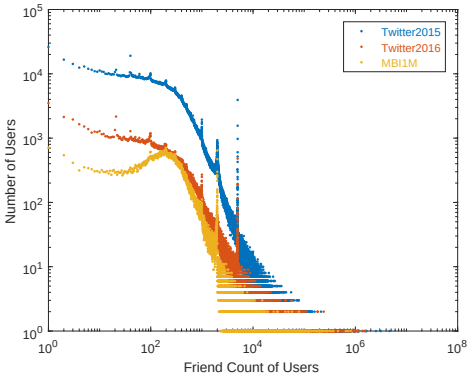
(a) Retweet distributions.



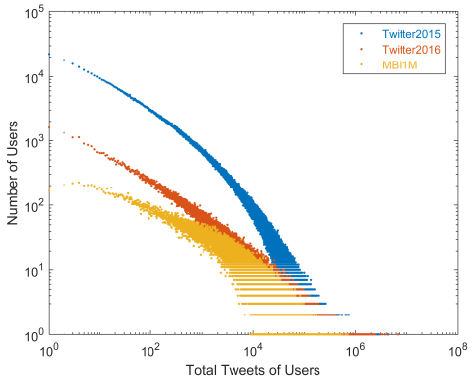
(b) Loglog plot of the user favorite tweet number.



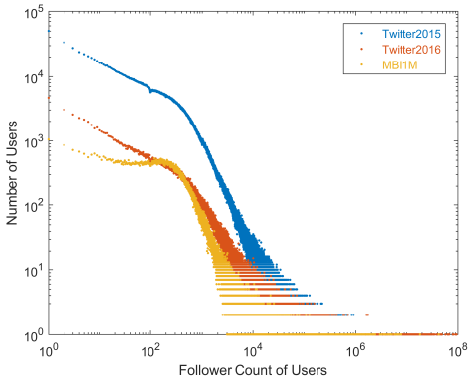
(c) User account history measured in years since account creation.



(d) Loglog plot of the user friend number.

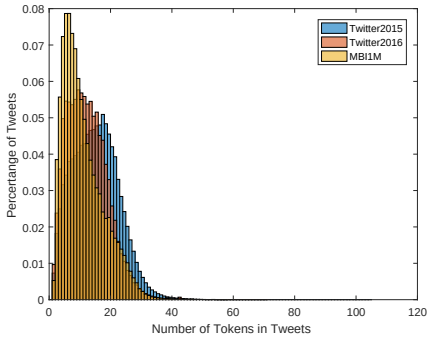


(e) Loglog plot of the user favorite tweet number.

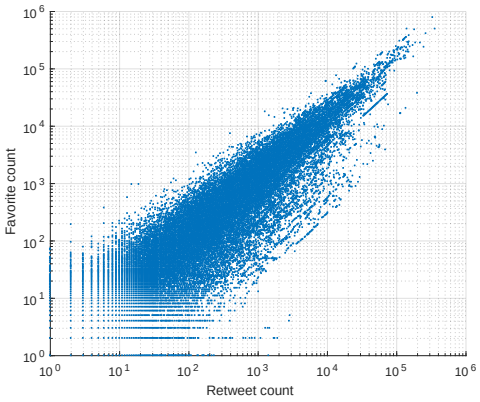


(f) Loglog plot of the user follower number.

Figure 2: Visualization on datasets.



(a) Tweet length distribution.



(b) Scatter plot for the retweet count and favorite count for a random subset of Twitter2016 dataset. This figures exhibits strong correlation between retweet counts and favorite counts.

(c) Word cloud with stopwords.

(d) Word cloud without stopwords.

Figure 3: More visualization on datasets.

Twitter user can *tweet* a new message, *favorite* one tweet, or *retweet* tweets shown on one's timeline. We accumulated the favorite and status update count (including original tweets and retweets) from three datasets and visualized them in Figure 2b and Figure 2e respectively. Similarly, the favorite and status count number also obey the "power-law" distribution.

We further studied the tweet level characteristics of the three datasets. Specially, we care about the retweet number distributions, tweet length, and the word used on Twitter. We show the retweet count of tweets of the three datasets in Figure 2a.

In most of the loglog plots, curve of the Twitter2015 dataset is clearly above the Twitter2016 dataset and the MBI1M dataset. One major reason for this is because Twitter2015 contains more tweets and users than the other two datasets. Thus the absolute number/count is relatively larger than the other two. However, all three datasets all clearly demonstrated the "power-law" distribution.

We also show the tweet token histogram in Figure 3a. Interestingly, the Twitter2015 and Twitter2016 datasets contains longer sentences, compared to MBI1M dataset. But in general, Twitter's 140 character limits make sentences relatively short. The mode of the three datasets are all less than 25 words per tweet.

We first built vocabulary from Tweet text and then rendered word clouds based on the word frequencies in the Tweet corpus. Notice that Twitter specific stopwords (like URLS, hashtags, words in all-caps) are very frequently used in tweets.

Table 3: Comparison of different popularity metric. Spearman: higher is better. MAPE: lower is better.

Output	Spearman			MAPE		
	MBI1M	T2015	T2016	MBI1M	T2015	T2016
Retweet count	0.229	0.358	0.350	0.057	0.084	0.103
Fav count	0.231	0.361	0.345	0.061	0.079	0.110

4.3 Favorite or Retweet Count?

Retweet count as a metric for predicting the popularity/virality is well recognized. We follow many other related work [5, 16, 20, 21] to use retweet count as the prediction output.

In addition to retweet count, the number of favorites might be another possible measure for evaluating content popularity. We use the same model as proposed in our paper, but replace retweet count with favorite count as the prediction output. Results are shown in Table 3.

As shown by Table 3, using favorite count and retweet count gives similar results on multiple datasets. We contribute such results to the strong correlation between the favorite counts and the retweet counts. To visualize such relationship, we randomly sampled 10k tweets from the Twitter 2016 dataset and visualize the relationship between the retweet count and the favorite count in Figure 3b. Such correlation can be easily inferred from Figure 3b.

4.4 Tweet Language Preprocessing

We listed the rules for tweet text preprocessing in Table 4.

Table 4: 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

4.5 Data Filtering

Twitter data are notoriously noisy and dirty. We also noticed the noise in tweet data. We used the Yahoo not safe for work (NSFW) CNN model to classify inappropriate tweets[1] in the Twitter2015 dataset. We investigated the influence of inappropriate contents in different settings: porn free, porn only, and mixed data. A porn-only set of 25K tweets is selected with scores no smaller than 0.8; a porn-free set of 25K tweets is select with scores no larger than 0.2. The Twitter2015 dataset is used as the mixed dataset.

We train two models on the porn-free and porn-only datasets from scratch. We report the results in Table 5.

Our method achieved similar performance across all datasets. Thus eliminating inappropriate data does not contribute more to our model. Our method can achieve good performance on different mixtures of appropriate and NSFW data because 1) the model capacity is large enough to capture the characteristics of both clean and dirty data; 2) retweet counts on the Twitter dataset exhibit the "power-law": most tweets have 0 or low retweet counts. Porn related tweets also obey this rule. Less retweeted data points do not contribute much to the loss function. There exists a relatively small

[1] https://github.com/yahoo/open_nsfw

Table 5: Data filtering results.

Dataset	Twitter2015	Twitter2016	Porn-Free	Porn-Only
Spearman	0.358	0.350	0.355	0.354
MAPE	0.084	0.103	0.095	0.097

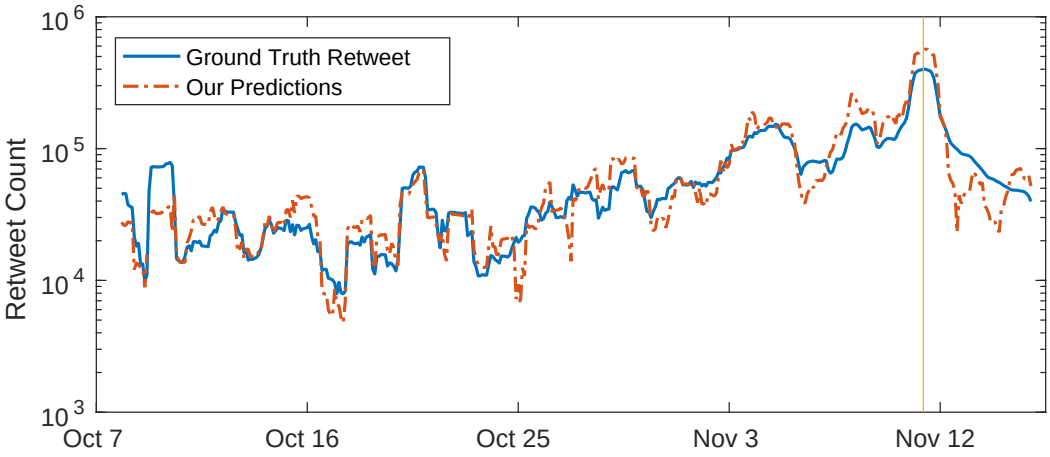


Figure 4: Tweet popularity during 2016 US presidential election.

number of highly retweeted NSFW tweets that can lead to large gradients of the loss function. Thus the model overall is not heavily impacted by NSFW data. This explains the results that filtering porn tweets does not make a big difference.

The existence of large amounts of NSFW data in Tweet streams explains that sexually related contents are likely to go viral. Modeling the data as a whole, without filtering inappropriate data, actually better represents the real world scenario.

5 Examples

5.1 US Election Example

Twitter acted as an important media in the US presidential election in 2016. We demonstrated that our model can catch the general popular trend during the election process. We collected tweets containing relevant keywords (*"president", "vote", "election", "Clinton", "Trump"*) from October 8, 2016 to November 14, 2016. In Figure 4, 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.

5.2 Positive and Negative Tweet Examples

We show positive output examples of our retweet prediction model in Figure 5 and Figure 6, and negative examples in Figure 7. Empirically, we found that it's easier to predict popularity for "concrete" concepts than predicting "abstract" concepts. For example, in Figure 7, many "abstract" concepts expressed by textual description are hard to be correlated to visual images. Such confusion should be further addressed to make the prediction more accurate.

References

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

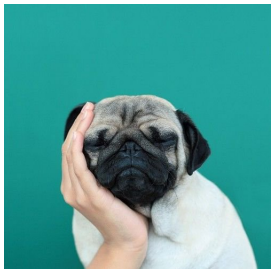
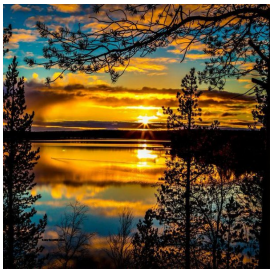
Image			
Tweet	Gotta admire these next-level pumpkin carving skills. Happy Halloween everyone! https://t.co/oFYg8Asnkb	God got me here! #may7 #23withadoctoratedegree https://t.co/rXjUdV9EXJ	Canadians are so careless with their money https://t.co/HNeDpEF7aD
Truth	9415	9424	9439
Prediction	8855	10038	9301
Image			
Tweet	pugs are drugs https://t.co/KPHthpXElI	@babygry Good afternoon, Dear https://t.co/MCRgw7U2mz	PSA: $1 CHRISTMAS SOCKS AT TARGET ARE A THING NOW https://t.co/7u8podCAum
Truth	9494	9496	11123
Prediction	9059	9990	11137
Image			
Tweet	Dig down deep inside of you The sun comes shining through It's all about love.. https://t.co/9zmmJVnria	Birthday cake pic for u. http://t.co/eyRACm0XLN	Ceraunophile (n.) - a person who loves lightning and thunder http://t.co/qnIZrlS2tR
Truth	11172	11353	37958
Prediction	14911	12967	25783

Figure 5: Positive examples tweets and their predictions. Our joint model correctly correlated the visual image and the textual description to produce accurate retweet number.

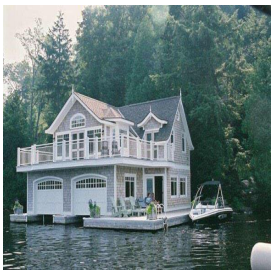
Image			
Tweet	paolo sebastian dresses are simply gorgeous https://t.co/tymvgRcwzQ	in case you've ever wondered what a space shuttle launch looks like at night https://t.co/CO2DeRP6C6	Sunflower field near Mount Fuji on Honshu Island, Japan by Shinichiro Saka https://t.co/y4mmObggTM
Truth	11387	14958	15033
Prediction	13120	16630	15258
Image			
Tweet	I want to participate in a sky lantern festival! https://t.co/bZE0uNvxfz	when you're a ******* mess but he loves and takes care of you anyway https://t.co/PkcrS7omMx	lake house goalss http://t.co/CM4vVGezm9
Truth	41520	32217	47279
Prediction	39961	33890	41950

Figure 6: Positive example of tweets and their predictions.

[2] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep canonical correlation analysis. In *ICML*, 2013. 2

[3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015. 2

[4] Hila Becker, Mor Naaman, and Luis Gravano. Beyond trending topics: Real-world event identification on twitter. *ICWSM*, 2011. 2

[5] Spencer Cappallo, Thomas Mensink, and Cees G.M. Snoek. Latent factors of visual popularity prediction. In *International Conference on Multimedia Retrieval*, 2015. 2, 5, 8

[6] Justin Cheng, Lada Adamic, P. Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. Can cascades be predicted? In *WWW*, 2014. 2

[7] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv 1412.3555*, 2014. 2

[8] Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *RANLP*, 2013. 2

[9] Arturo Deza and Devi Parikh. Understanding image virality. In *CVPR*, 2015. 2

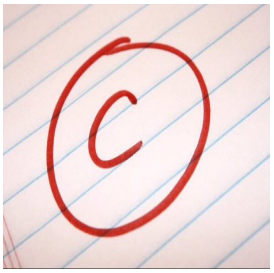
Image			
Tweet	this is what I want and need https://t.co/lIgWTGLe7X	this is the only upgrade I care about right https://t.co/h1FXSTJRxe	Good morning all twitter friends https://t.co/h1kWkpH7pU
Truth Prediction	15213 7556	11317 2894	11359 3197
Image			
Tweet	Pluviophile http://t.co/ARTixuHBys	serenity https://t.co/2yj7iGJMrd	It's all perspective. https://t.co/OdhgDDxxvG
Truth Prediction	35627 1892	36225 2781	38193 4592
Image			
Tweet	this is the best thing you'll see all day https://t.co/G7Gp4eGPRO	Wow! It's perfectly timed http://t.co/WpArlrlU1b	God is an artist https://t.co/tMMcsIdG2W
Truth Prediction	11204 587	11280 4318	32382 2870

Figure 7: Negative example of tweets and their predictions. Our joint model failed to understand "abstract" or high level concepts expressed by textual descriptions or visual images, thus failed to correlate different modalities, giving inaccurate estimates.

[10] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc\textquotesingle Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 2

[11] Haoyuan Gao, Junhua Mao, Jie Zhou, Zhiheng Huang, Lei Wang, and Wei Xu. Are you talking to a machine? dataset and methods for multilingual image question answering. In *NIPS*, 2015. 2

[12] Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009. 2

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

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

[15] Akshay Java, Xiaodan Song, Tim Finin, and Belle 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, 5

[16] Aditya Khosla, Atish Das Sarma, and Raffay Hamid. What makes an image popular? In *WWW*, 2014. 2, 8

[17] Ryota Kobayashi and Renaud Lambiotte. Tideh: Time-dependent hawkes process for predicting retweet dynamics. *arXiv:1603.09449*, 2016. 3, 5

[18] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *WWW*, 2010. 2

[19] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *ECCV*, 2016. 2

[20] Masoud Mazloom, Robert Rietveld, Stevan Rudinac, Marcel Worring, and Willemijn van Dolen. Multimodal popularity prediction of brand-related social media posts. In *ACM MM*, 2016. 8

[21] Philip J. McParlane, Yashar Moshfeghi, and Joemon M. Jose. "nobody comes here anymore, it's too crowded"; predicting image popularity on flickr. In *ICMR*, 2014. 2, 8

[22] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernocký, and Sanjeev Khudanpur. Recurrent neural network based language model. *Interspeech*, 2010. 2

[23] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv 1301.3781*, 2013. 2

[24] Jiquan Ngiam, Aditya Khosla, Mingyu Kim, Juhan Nam, Honglak Lee, and Andrew Y. Ng. Multimodal deep learning. In *ICML*, 2011. 2

[25] Evan Shelhamer, Jonathan Long, and Trevor Darrell. Fully convolutional networks for semantic segmentation. *IEEE TPAMI*, 2016. 2

[26] Hua-Wei Shen, Dashun Wang, Chaoming Song, and Albert-László Barabási. Modeling and predicting popularity dynamics via reinforced poisson processes. *arXiv:1401.0778*, 2014. 3

[27] Richard Socher, Cliff C Lin, Chris Manning, and Andrew Y Ng. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, 2011. 2

[28] Richard Socher, Andrej Karpathy, Quoc V. Le, Chris D. Manning, and Andrew Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2013. 2

[29] Nitish Srivastava and Ruslan R Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*. Curran Associates, Inc., 2012. 2

[30] Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In *NIPS*, 2014. 2

[31] Chenhao Tan, Lillian Lee, and Bo Pang. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *ACL*, 2014. 2

[32] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE TPAMI*, 2016. 2

[33] Liwei Wang, Yin Li, and Svetlana Lazebnik. Learning deep structure-preserving image-text embeddings. In *CVPR*, 2016. 2

[34] Kota Yamaguchi, Tamara L Berg, and Luis E Ortiz. Chic or social: Visual popularity analysis in online fashion networks. In *ACM MM*, 2014. 2

[35] Qingyuan Zhao, Murat A. Erdogdu, Hera Y. He, Anand Rajaraman, and Jure Leskovec. Seismic: A self-exciting point process model for predicting tweet popularity. In *KDD*, 2015. 2

[36] Changtao Zhong, Dmytro Karamshuk, and Nishanth Sastry. Predicting pinterest: Automating a distributed human computation. In *WWW*, 2015. 2