Deep Exogenous and Endogenous Influence Combination for Social Chatter Intensity Prediction

[1]Subhabata Dutta, [2]Sarah Masud, [3]Soumen Chakrabarti, [2]Tanmoy Chakraborty
[1]Jadavpur University, India; [2] IIIT-Delhi, India; [3] IIT Bombay, India

ABSTRACT

Modeling user engagement dynamics on social media has compelling applications in market trend analysis, user-persona detection, and political discourse mining. Most existing approaches depend heavily on knowledge of the underlying user network. However, a large number of discussions happen on platforms that either lack any reliable social network (news portal, blogs, Buzzfeed) or reveal only partially the inter-user ties (Reddit, Stackoverflow). Many approaches require observing a discussion for some considerable period before they can make useful predictions. In real-time streaming scenarios, observations incur costs. Lastly, most models do not capture complex interactions between exogenous events (such as news articles published externally) and in-network effects (such as follow-up discussions on Reddit) to determine engagement levels.

To address the three limitations noted above, we propose a novel framework, `ChatterNet`, which, to our knowledge, is the first that can model and predict user engagement *without considering the underlying user network*. Given streams of timestamped news articles and discussions, the task is to observe the streams for a short period leading up to a time horizon, then predict *chatter*: the volume of discussions through a specified period after the horizon. `ChatterNet` processes text from news and discussions using a novel time-evolving recurrent network architecture that captures both temporal properties within news and discussions, as well as influence of news on discussions. We report on extensive experiments using a two-month-long discussion corpus of Reddit, and a contemporaneous corpus of online news articles from the Common Crawl. `ChatterNet` shows considerable improvements beyond recent state-of-the-art models of engagement prediction. Detailed studies controlling observation and prediction windows, over 43 different subreddits, yield further useful insights.

ACM Reference Format:
[1]Subhabata Dutta, [2]Sarah Masud, [3]Soumen Chakrabarti, [2]Tanmoy Chakraborty. 2020. Deep Exogenous and Endogenous Influence Combination for Social Chatter Intensity Prediction. In *Proceedings of the 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '20), August 23–27, 2020, Virtual Event, CA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394486.3403251

KDD '20, August 23–27, 2020, Virtual Event, CA, USA

ACM ISBN 978-1-4503-7998-4/20/08…$15.00
https://doi.org/10.1145/3394486.3403251

1 INTRODUCTION

The Web is the most popular medium for large-scale public interaction and information propagation. About 4.3 billion people used the Internet in 2019, with 3.53 billion using at least one social media site[1]. Unlike radio or television, social media convey information with active participation of users. One can broadly identify two modes of engagement within user communities. In the *reshare* mode, a user shares some information with a community (friends, followers, groups, etc.), and members of that community recursively propagate the information. This process creates a tree of reshares, where information flows from the root to the leaves. The other is the *reply* mode, where one user posts some opinion, and other users reply to that post (or to other replies of the post), thus, forming a discussion. Mining the dynamics of these modes can yield useful insights for opinion mining, market research [11, 19], political analysis [8, 12] and human psychology [14].

A growing body of research has focused on modeling the dynamics of such information propagation — both for *reshare* [17, 24, 28] and *reply* [21]. There are broadly two approaches. *Feature-driven models* mainly rely on three types of features for modeling the growth of reply trees, based on the social network among users, the propagated content, and temporal observations. The other approach is to fit *self-exciting process* models [27] (reviewed in Section 5). In terms of growth prediction, the existing body of literature has a bias for reshare trees; specifically, Twitter retweet trees.

Latent/Implicit social networks. Most reshare models depend heavily on the user network (*follow* on Twitter, *friend* on Facebook), which is available on only a limited number of platforms. However, if we focus on the dynamics of *discussions*, many platforms do not offer explicit user-user social ties. One such influential platform is Reddit. As of 2019-12-04, it is used by 430 million active monthly users[2]. Users can post *submissions* to one of the Reddit communities, commonly called *subreddits*. Other users can then *comment* in reply to the submission or any earlier comment on the submission. These comments form a discussion tree, with the submission at the root. Though Reddit provides a *subscribe* option to its users, using which they implement an internal community structure (and not an inter-user link as in Twitter or Facebook), Reddit keeps this information private. HackerNews, IRC, and Slack assert similar constraints. Engagement prediction methods that rely on the social network structure will lose applicability and performance.

Exogenous effects. A second critical modeling issue is *exogenous* influence. An influential event (recurring or sporadic) in the real world determines what topic will be the "talk-of-the-town" on the social media. In Figure 1, we show how the activities in Reddit

[1]Source: https://www.statista.com/
[2]https://www.adweek.com/digital/reddit-reaches-430-million-monthly-active-users-looks-back-at-2019/

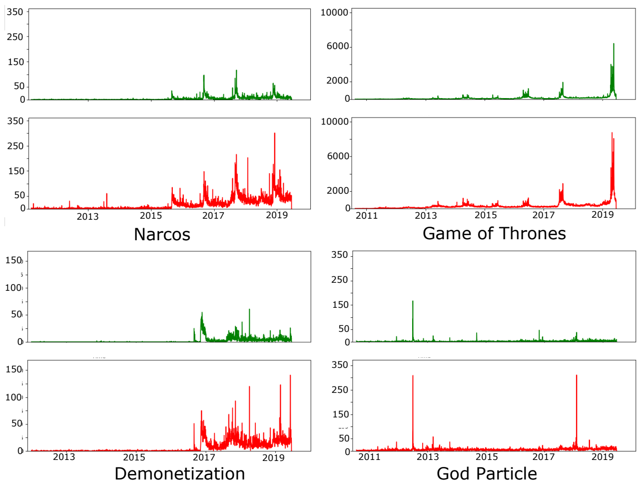

Figure 1: Plots showing Reddit submissions (in green) and comments (in red) posted per hour related to four different keywords, changing over time. Y-axis in all eight plots represent absolute counts. Each keyword corresponds to a different exogenous event. We can observe spikes in the Reddit activity coinciding with the event timings.

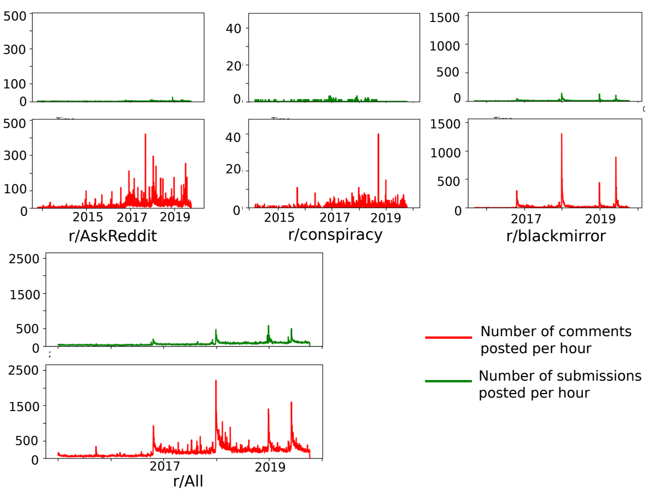

Figure 2: Hourly submission and comment count for the keyword *Black Mirror* (a popular Web-series). We show the counts in three different communities (subreddits), and across whole of Reddit. We can clearly observe the differences in user reaction towards the same event in different communities.

change over time for four different topics, each corresponding to different events. *Narcos* and *Game of Thrones* are popular television series, releasing each season (collection of episodes) periodically. For both of these recurring events, there are sudden spikes in the number of submissions and comments. The last spikes in activities correspond to the final seasons where we can observe a huge gap in the number of submissions and comments, corresponding to more extensive discussions (high comment per submission ratio). The discovery of Higgs Boson (2012) and its decay (2018) corresponds to the two abrupt spikes in the activity plots of *God Particle*. *Demonetization* was an event of national importance in India in November 2016. Owing to its long-standing effects and several events triggered by it, we can observe multiple spikes even after the initial one. It was intensely discussed during the parliamentary elections of India in 2019, as reflected in the large spikes with a high comment-to-submission ratio in the first quarter of 2019. *Demonetization* is a perfect example of the superposition of multiple exogenous events, financial and political. From these examples, it is quite evident that a model that takes these exogenous events into consideration during training may result in better user engagement predictions.

Endogenous (within-community) influence. While external events play a crucial role in determining "what people will talk about", the degree to which such 'chatter' will evolve, and the time it will take to decay, are strongly dependent on the internal or *endogenous* states of a community. Figure 2 shows activity levels related to *Black Mirror* in different subreddits. Whereas the whole of Reddit (*r/all*) and the dedicated subreddit *r/blackmirror* follow the recurring pattern of event arrival (release of seasons), this is not the case for *r/AskReddit* and *r/conspiracy*. Interestingly, the subreddit *r/blackmirror* was created at the end of 2016. Flocking of dedicated followers to this subreddit and their intense engagement played a pivotal role in raising the topic's popularity in other subreddits and in Reddit overall (sharp spikes in Figure 2 after 2017 compared to the previous releases in 2014–15).

`ChatterNet`. We present `ChatterNet`, a system for chatter prediction that handles the combined challenges of unknown influence network structure and exogenous influence. `ChatterNet` observes news and discussion streams for a limited time window up to a time **horizon**, after which it predicts **chatter**: the intensity of subsequent discussion up to another specified time.

(For concreteness, throughout this paper, we will use **news** articles as the prototypical exogenous influence. We will use **submissions**, **comments**, and **discussion** as prototypical social network activity. Note that these are broad model concepts that may be embodied differently in other chatter prediction applications.)

`ChatterNet` achieves our goal using a network architecture inspired by a **unified chatter model**. In this model, each user follows a two step process of *read and react*. Upon the arrival of a discussion item, a user reads its content (or views images or video). Then, depending on his/her cognitive state and the content features (topic, complexity, opinion), s/he decides whether or not to react and contribute to the discussion. Any contribution, in turn, affects the state of other users. This process, "read and react", aggregated over uesrs, can be conceptualized as an evolving mapping of content to its virality, conditioned on the dynamics of the exogenous and the endogenous states.

Using two months of Reddit discussions on 43 different subreddits, amounting to nine million submissions and comments, along with 3.9 million time-aligned news articles, we show that `ChatterNet` makes more accurate chatter predictions compared to recent competitive approaches based on Hawkes Processes [17], cascades [4], and others. Drilling down into the subreddits and contrastic their dynamics give additional insights.

Summary of contributions. Our contributions are four-fold:

- Formal specification of a new chatter prediction problem in settings where social network knowledge is absent and exogenous influence is present.
- Design and implementation of ChatterNet, a new chatter prediction system that targets the above setting.
- Extensive experimental comparison against prior chatter prediction methods, demonstrating the superiority of ChatterNet.
- New chatter prediction data set and accompanying code[3].

2 DESIGN OF CHATTERNET

Guided by the unified chatter model discussed in Section 1, we describe in this section the complete working of ChatterNet. We set up notation in Section 2.1, and discuss some basic approaches that cannot capture all the signals available in our setting. We describe how ChatterNet combines these signals from news and discussion, in Section 2.2. Then, in Section 2.3 we motivate why events arriving after the horizon need time-evolving network components, and present a suitable network for processing post-horizon events. We tie these pieces together with a training loss in Section 2.4. Figure 3 shows a sketch of ChatterNet.

2.1 Preliminaries and Notation

Time is quantized into observation intervals of length Δ_{obs} (e.g., $[t_{k-1}, t_k]$ in Figure 3). Intervals are indexed with k. We denote as $\hat{N} := \langle (n_i, t_i) | \forall i \in \mathbb{Z}^+ \rangle$ the stream of news articles n_i with publication timestamps t_i. Let $\hat{S} := \langle (s_j, t_j) | \forall j \in \mathbb{Z}^+ \rangle$ be a stream of submission items s_j posted at time t_j. Every news item n_i consists of a text digest with headline and body, while every submission item s_j is a triplet (s_j^T, s_j^V, s_j^R), where s_j^T is the text digest of s_j, s_j^V is the subreddit to which s_j was posted, and s_j^R is the average commenting activity (number of comments) in the subreddit within the previous interval. For any submission s_j posted at timestamp t_j with $t_k < t_j \leq t_{k+1}$, we define an **observation window** $[t_j, t_j + m\Delta_{\text{obs}}]$ and a **prediction window** $[t_j + m\Delta_{\text{obs}}, t_j + \Delta_{\text{pred}}]$, where $m \in \mathbb{N}$ is an application-driven hyperparameter. Note that, in this setting it is important to differentiate the roles of submissions posted before and after t_k, which is the boundary up to which we have the most recent exogenous-endogenous signals defined. So every submission up to t_k contributes to this endogenous signal. The submission posted within $t_k < t_j \leq t_{k+1}$ lies between the two arrivals of influence signal, one in the past (t_k) and one in the future (t_{k+1}). So, every such submission might be under the influence signal at t_k.

Our goal is to predict **chatter** pertaining to s_j, defined as $y_j = \ln(1 + C_j)$, where C_j is the total number of comments made about s_j within the prediction window, after observing commenting activity about s_j within the observation window. (Additive error in log-count prediction amounts to count prediction within a multiplicative factor. C_j depends on Δ_{pred} but we elide that for simpler notation.) As early predictions are most beneficial, we specify two settings:

Zero-shot: Empty observation window, with $m = 0$.
Minimal early observation: Here $m > 0$, but $m\Delta_{\text{obs}} \ll \Delta_{\text{pred}}$.
Popular time-series models cannot perform the above tasks well, for several reasons. First, any static mapping from the textual features of the submissions to their corresponding future chatter fails to incorporate the dynamic exogenous and endogenous influences that govern chatter. Second, generative models need a substantial degree of early observation which is not available in our setting. Third, the high arrival rates demand fast response. Therefore, when predicting the future chatter of a submission, chatter under its predecessors in the stream remains mostly unobservable. This precludes autoregressive [17] approaches. Finally, the lack of knowledge of user-user ties inhibits the employment of information diffusion models [4].

The cumulative aggregate influence of exogenous and endogenous signals from news and submission streams during $[t_{k-1}, t_k]$ will be endowed a deep representation G_k, as defined in Section 2.2. G_k will help map submission texts posted within the next interval to a base chatter intensity, which will be aggregated with the activity within the observation window to predict the final chatter.

In what follows, every weight and bias matrix (denoted as W and Q, respectively with various subscripts) belongs to the trainable model parameter set. Every text segment (news and submission) is mapped to a sequence of low dimensional representations using a shared word embedding layer. Moreover, the subreddit information s_j^V corresponding to each submission s_j is mapped to a **subreddit embedding vector** $\mathbf{U}_j$ using a shared embedding layer. All these embeddings are part of the trainable parameters of ChatterNet.

2.2 Cumulative Influence Aggregation

We define two functions that compute exogenous and endogenous influences:

$$\text{Exogenous:} \quad \mathcal{F}_{\text{X}}(N | t_i \leq t \; \forall (n_i, t_i) \in N \subseteq \hat{N}) \tag{1}$$

$$\text{Endogenous:} \quad \mathcal{F}_{\text{E}}(S | t_j \leq t \; \forall (s_j, t_j) \in S \subseteq \hat{S}) \tag{2}$$

We model each of these functions $\mathcal{F}_{\text{X}}$ and $\mathcal{F}_{\text{E}}$ in two steps.

First, we map each text digest to its influence feature map using a shared convolution block (refer to component (1) in Figure 3). Given an input sequence of word vectors of a news (submission text) as n_i (s_j^T), we apply successive 1-dimensional convolution and max-pooling operations to produce a feature map X_i^n (X_j^s):

$$\begin{aligned} C_i^n &= \text{ReLU}\big(\text{Conv1D}(n_i | W_{\text{static}})\big) \\ X_i^n &= \text{MaxPool}(C_i^n) \end{aligned} \tag{3}$$

where W_{static} denotes filer kernels and C_i^n (C_j^s) is an intermediate representation. We use parallel branches of convolution and pooling operations with different kernel sizes (1, 3, and 5) to capture textual features expressed by contexts of different sizes. The outputs from each branch are then concatenated to produce the final feature representation f_i^n (f_j^s) corresponding to n_i (s_j) (detailed organization explained in the Appendix, Figure 5). The submission feature maps are concatenated with subreddit vector $\mathbf{U}_j$ to differentiate the influences of submissions from different subreddits, as discussed in Section 1. The final feature map corresponding to s_j is then f_j^{sv}. We denote these stages as *static* because W_{static} remains temporally invariant.

ConvNet vs. LSTM/BERT. We chose a simple convolutional architecture [13, 15] over a recurrent one for two reasons — i) convolution reduces the size of parameter space, which is essential in handling large data streams, and ii) our task requires efficient understanding of topic-specific keywords to constitute the influence, as opposed to the complex linguistic structures with long-term

[3]Code and Sample Data at https://github.com/LCS2-IIITD/ChatterNet

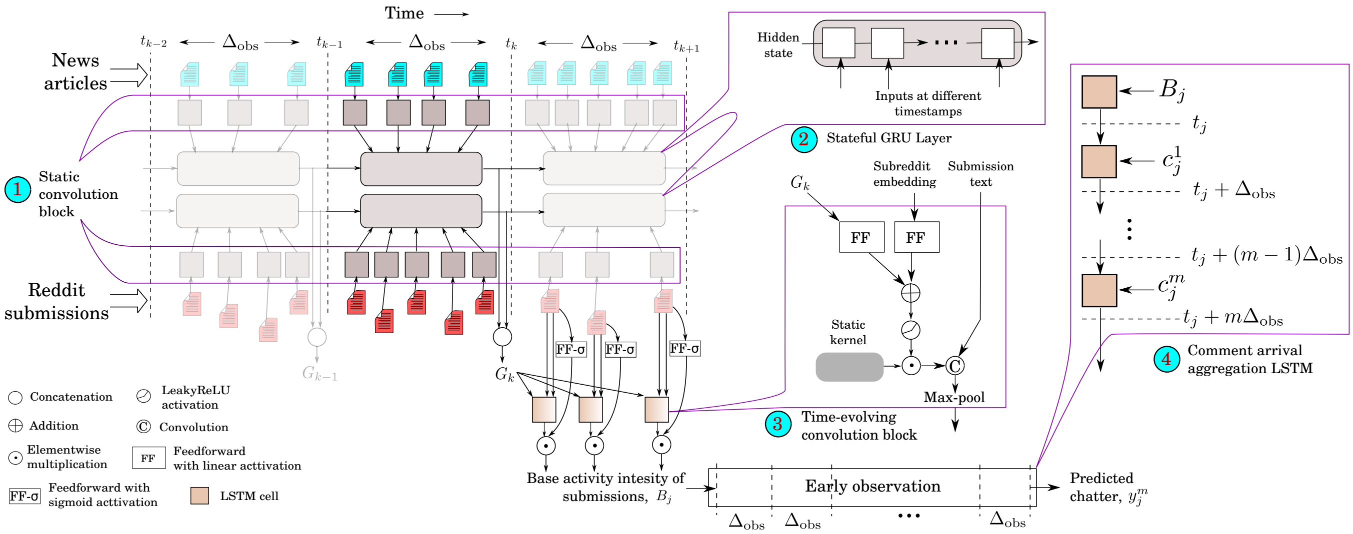

Figure 3: Steps of combining exo- and endogenous influence to predict future chatter submissions over time. Indices j and k correspond to submissions and intervals, respectively. (i) Features of news articles (exogenous) and submissions (endogenous), published within interval $[t_{k-1}, t_k]$ of length Δ_{obs}, are extracted using a static convolution block. (ii) Two GRUs with separate parameters aggregate each set of feature maps over the interval, and their final hidden states are concatenated to G_k, called the *time-evolving influence state.* (iii) Shared instances of a time-evolving convolution block extract features of submissions made in $[t_k, t_{k+1}]$, controlled by G_k and the subreddit embedding of the submissions. Every submission also includes average commenting rate in the corresponding subreddit, which is passed through a feedforward layer with sigmoid activation and multiplied with the outputs of time-evolving convolutions to compute base activity intensity B_j of each submission. Finally, (iv) the number of comments during each of m next time intervals is aggregated with B_j to predict future chatter y_j^m.

dependencies; this makes recurrent architectures an overkill. Our experiments with BERT [6] instead of convolution to produce text representations (expectedly) gave no significant gain.

Next, using the convoluted feature maps f_i^n (f_j^{sv}), we compute discrete approximations of the functions $\mathcal{F}_{\text{X}}$ ($\mathcal{F}_{\text{E}}$) at the end of every interval $[t_{k-1}, t_k]$ as G_k^n (G_k^s), as follows (see component (4) in Figure 3):

$$G_k^n = \text{GRU}\Big(h_{k-1}^n, \langle f_i^n | t_{k-1} < t_i \le t_k \rangle | W_G^n\Big) \tag{4}$$

where GRU is a Gated Recurrent Unit, t_i (t_j) corresponds to the timestamp associated with n_i (s_j), h_{k-1}^n is the hidden state of the GRU from the previous interval, and W_G^n (W_G^s) is the parameter set. Stateful propagation of the hidden state ensures the modeling of short-term as well as long-term influence signals. Finally, the cumulative influence G_k is computed as the concatenation of G_k^n and G_k^s.

We deploy two different GRUs to aggregate the exogenous and endogenous influences over the intervals given the different arrival patterns of news articles over web and submissions over Reddit (news come in sparse bursts, while submissions mostly come in very high rate).

Again, our choice of GRU as the recurrent information processing layer for this task is motivated by our experiments confirming LSTMs to be slower with no performance gain compared to GRUs.

2.3 Time-evolving Convolution (TEC)

Following the intuitive motivation of the unified chatter model, we may now seek to map any submission s_j posted in the interval $[t_k, t_{k+1}]$ to its potential to invoke future chatter, controlled by the cumulative influence from the previous interval. Formally, this mapping can be defined as,

$$\mathcal{B} = \mathcal{F}_G(s_j | G_k,\ t_k < t_j \le t_{k+1}) \tag{5}$$

We again resort to 1-dimensional convolution to learn feature maps from s_j, but this time, the filter kernel W_{TEC} being a function of G_k and the subreddit vector $\mathbf{U}_j$ corresponding to s_j:

$$W_{\text{TEC}} = W_S \odot \gamma(W_G \cdot G_k + W_V \cdot s_j^V) \tag{6}$$

where $\gamma(x) = x$ if $x \ge 0$, and αx of $x < 0$ (standard LeakyReLU activation with parameter α, experimentally set to 0.2). W_G and W_V (corresponding to the two feed-forward layers inside component (3) in Figure 3) control the contributions of the cumulative influence and the subreddit, respectively. This $\gamma(W_G \cdot G_k + W_V \cdot s_j^V)$ component 'calibrates' the static kernel W_S with the element-wise multiplication according to the influence. As G_k evolves over time, so does W_{TEC}.

Equipped with this influence-controlled kernel, the time-evolving convolution and max-pooling on s_j can be defined similar to Eq. 3. Again, we apply parallel branches of successive convolution with different filter sizes and max-pooling, concatenation, and another series of convolutions (complete organization shown in Appendix, Figure 6) to finally map s_j to a non-negative real value $\tilde{B}_j$, the **potential chatter intensity** of s_j independent of the subreddit where s_j is posted.

2.4 Final Prediction

Chatter levels in different subreddits vary with the number of active users at any time. The average commenting activity s_j^R of the subreddit corresponding to s_j enables us to compute the relative activity signal $r_j \in \mathbb{R}$ such that,

$$r_j = \sigma(W_R \cdot s_j^R + Q_R) \tag{7}$$

where $\sigma(x) = (1 + e^{-x})^{-1}$ (shown as "FF-σ" blocks in Figure 3). We compute the **base chatter intensity** corresponding to s_j as $B_j = r_j \tilde{B}_j$ such that $0 < r_j < 1$ plays the role of a scaling factor to calibrate B_j according to the activity level of the subreddit.

Having computed B_j, the chatter intensity invoked by s_j, under the influence of past history of news and submission arrival and calibrated by the subreddit information, we next observe the commenting activity under s_j within the observation window (see component (4) in Figure 3). We employ a binning over time intervals to transform the comment arrivals within the observation window $[t_j, t_j + m\Delta_{\text{obs}}]$ into a sequence $\langle c_j^1, c_j^2, \cdots, c_j^m \rangle$ where each c_j^l is the total number of comments arrived within $[t_j + (l-1)\Delta_{\text{obs}}, t_j + l\Delta_{\text{obs}}]$. This sequence serves as a coarse approximation of the rate of comment arrivals over the observation window. We use a single LSTM layer to aggregate this sequence and predict the final chatter y_j^m (superscript corresponds to the length of the observation window). In the zero-shot setting (i.e., $m = 0$) this LSTM is not used and we predict chatter $y_j^0 = B_j$.

Details of `ChatterNet` parameters are given in Appendix B.

2.5 Cost/Loss Functions

In a realistic setting, there are far too many discussions invoking near-zero chatter along with very small number of those which go viral heavily. As we take both of these types without any filtering (opposed to excluding less viral ones in the cascade prediction tasks like [17] or [27]), the cost function needs to handle skewed ground truth values. To deal with this, we train `ChatterNet` by minimizing the mean absolute relative error given by $\sum_j \frac{|y_j - y_j^m|}{y_j + \epsilon}$, where y_j is the chatter ground truth, y_j^m is the predicted chatter, and ϵ is a small positive real number to avoid division by zero (as implemented in Keras/Tensorflow).

3 EXPERIMENTS

3.1 Dataset

We collected the discussion data from Pushshift.io[4], a publicly available dump of Reddit data, stored in monthly order. We used the discussion data of October and November 2018, from 43 different communities (Subreddits). The October data was used for training and development, and the November data was used for testing. In total, we have a collection of 751,866 submissions with 2,604,839 comments in the training data, and 1,334,341 submissions with 4,264,177 comments in the test data.

To fetch the news articles published online, we relied on the news-please crawler [10], which extracts news articles from the Common Crawl archives[5]. We crawled the news articles published in the same timeline as of the Reddit discussions. We got 1,851,022 articles from 4757 different news sources for the month of October, and 2,010,985 articles from 5054 sources for November. It should be noted that this covers all the news articles published in English in this period, as we chose not to use non-English data. Details of preprocessings are given in Appendix A.

[4]https://files.pushshift.io/reddit/

[5]https://commoncrawl.org/

Table 1: Feature set for the baseline *CasPred*, where T:= set of unique terms in corpus, tf_t:= term frequency of term $t \in T$ in the submission, $|w|$:= number of words in the submission, $|cw|$:= number of words in the submission with more than 6 letters, $|s|$:= number of sentences in the submission, k:= observable discussion size, t_i:= time when i-th comment was put, t_0:= time of submission. * signifies a feature not in the original paper but added by us.

Features	Expression
Bag-of-words *	Unigram features with tf-idf
Complexity *	Degree of unique tokens used, $p = \frac{1}{\lvert T \rvert} \sum_{t \in T} tf_t(\log \lvert T \rvert - \log(tf_t))$
LIX Score *	Readability score of the submission, $r = \frac{\lvert w \rvert}{\lvert s \rvert} + 100 \times \frac{\lvert cw \rvert}{\lvert w \rvert}$
Polarity	Sum of sentiment intensity scores of the unique terms of the submission, computed using SenticNet [3]
Referral count *	Number of URLs in the submission
Size *	Number of words and sentences in the submission
Subreddit *	In which subreddit the submission is posted
Commenting time	Time elapsed between the i-th comment and the submission
Average time difference in first $k/2$ comments	$\frac{1}{k/2-1} \sum_{i=1}^{k/2-1} (t_i - t_{i-1})$
Average time difference in last $k/2$ comments	$\frac{1}{k/2-1} \sum_{i=k/2}^{k} (t_i - t_0)$

3.2 Training and Evaluation Protocols

For training the model, we use the October data divided into the train-validate split. The news and discussion streams in the period from 2018-10-03 00:00 GMT to 2018-10-22 23:59 GMT are used for training `ChatterNet`, while the validation is done using the data from 2018-10-23 00:00 GMT to 2018-10-30 23:59 GMT. We set Δ_{obs} and Δ_{pred} to 60 seconds and 30 days, respectively. We train multiple variations of `ChatterNet` with different observation windows: 15, 30, 45, and 60 minutes ($m = 15, 30, 45, 60$). Additional training detail are given in the Appendix C.

3.3 Baseline Models

Due to the novelty of the problem setting and the absence of social network information, comparing `ChatterNet` with the state-of-the-art is not straightforward. We engage four external baselines for retweet cascade prediction and Reddit user engagement prediction, tailored to our setting. We also implement multiple variants of `ChatterNet` for extensive ablation analysis of the different signals.

3.3.1 ***TiDeH.*** To adapt Time Dependent Hawkes Process [17] as a baseline in the absence of any knowledge of the underlying user network, we set the follower count of each Reddit poster/commenter as 1. In addition, we set the minimum thread size (minimum number of comments) to 10.

3.3.2 ***CasPred.*** The (re-)sharing cascade prediction approach of Cheng et al. [4] provides an interesting baseline for `ChatterNet`

Table 2: Evaluation of `ChatterNet` and the external baselines (over complete test data). τ and ρ correspond to Kendall's τ and Spearman's ρ, respectively, whereas Step-wise τ corresponds to Kendall τ on the sampled ground truth. `ChatterNet+` and `ChatterNet++` correspond to the zero-shot and minimal early observation of 1 hour.

Model	MAPE	τ	ρ	Step-wise τ
`ChatterNet+`	33.142	0.4042	0.4601	0.8781
`ChatterNet++`	**25.893**	**0.4439**	**0.5050**	**0.8980**
TiDeH (1hr.)	35.178	0.0715	0.1140	0.5622
CasPred-full	-	-	-	0.4741
CasPred-org	-	-	-	0.3515
RGNet	148.34	0.1871	0.2273	0.5305
DeepCas	163.6	0.2362	0.3309	0.2636

by allowing us to test not only the temporal but also the textual features of our dataset. Due to limitations of Reddit metadata we can make use of only a subset of the features they used. We also include some additional content features fitting to discussions in Reddit [7] (complete feature set in Table 1). We implement CasPred-org (original features) and CasPred-full (augmented with additional features) with observable cascade size $k = 10$.

3.3.3 ***RGNet.*** Our third external baseline is an adaptation of the Relativistic Gravitational Network [7], primarily designed to predict user engagement behavior over Reddit.

3.3.4 ***DeepCas.*** DeepCas [20] makes use of the global weighted topology of inter-user ties. Since, Reddit does not posses any explicit user-user mapping, we consider an edge between the posters and the commenters of a post (to generate the global network). Also, each post(with its set of poster and commenters) is treated as a cascade.

3.3.5 ***Ablation variants of `ChatterNet`.*** To observe the contribution of different components of `ChatterNet`, we implement the following ablated variants:

- **`ChatterNet-N`** which uses only the news-side influence signal;
- **`ChatterNet-S`** which uses only the submission-side influence signal;
- **`ChatterNet-Static`** which does not use any influence signal; for this, the time-evolving convolution block is replaced by a static convolution block;
- **LSTM-CC** which uses only the LSTM layer aggregating the observed comment arrivals (Section 2.4). LSTM-CC allows implementation for only the minimal early observation setting; zero-shot is not supported. Other variants are implemented for both of the task settings.

4 EVALUATION

We explore multiple evaluation strategies to see how `ChatterNet` responds to different challenges of chatter prediction. We use three different evaluation metrics: a) Mean Absolute Percentage Error (MAPE), b) Kendall rank correlation coefficient (Kendall's τ), and c) Spearman's ρ. As the CasPred model does not predict the exact size of the discussion but gives a binary decision of whether a given submission will reach at least size $l \times k$, $l \in \mathbb{Z}^+$ after observing a growth of size k, we can not evaluate this with the mentioned three metrics directly. Instead, we map the ground-truth to these $l \times k$

Table 3: Performances of different ablation variants of `ChatterNet` (see Section 3.3.5). Except for LSTM-CC, all the variations are tested for zero-shot (ZS) and early observation of 1 hour (OBS). MAPE and τ are as mentioned in Table 2.

Model	Setting	MAPE	τ
`ChatterNet`	ZS	33.142	0.4042
	OBS	25.893	0.4439
`ChatterNet-N`	ZS	38.498	0.3267
	OBS	29.023	0.4134
`ChatterNet-S`	ZS	37.007	0.3511
	OBS	28.890	0.4201
`ChatterNet-Static`	ZS	195.344	0.1488
	OBS	149.558	0.3580
LSTM-CC	OBS	152.313	0.3580

values such that the label of a discussion with size d would be $\lfloor \frac{x}{k} \rfloor$. Then we compute Kendall's τ over this values (hereafter called as step-wise τ). We use the same binning (with $k = 10$) for rest of the models to evaluate the step-wise τ.

4.1 Overall Performance

In Table 2, we show the evaluation results for `ChatterNet` in zero-shot and early observation settings along with all the external baselines. While `ChatterNet` exploiting comment arrival within the early observation window outperforms rest of the models by a large margin, it also performs better than the external baselines in the zero-shot setting.

An interesting pattern can be observed with TiDeH, RGNet and DeepCas. While TiDeH produces predictions comparable to `ChatterNet` in terms of MAPE, it suffers largely in terms of rank correlation. On the other hand, both RGNet and DeepCas follow a completely opposite pattern – better ranking of future chatter compared to predicting the actual value of chatter.

The poor performance of CasPred and RGNet can be explained in terms of the difference between their original design context and the way they are deployed in our problem setting. Almost two-third of the feature set originally used for CasPred can not be implemented here. Also, it is evident that our additional feature set actually improves the performance of CasPred, signifying the importance of these features for engagement modeling in Reddit. In case of RGNet, it is built to take into account the dynamics of user engagement over time. But in the absence of a rich feature set and social network information, it is the use of endogenous and exogenous influence which gives `ChatterNet` such leverage compared to the baselines.

Earlier we highlighted the major challenge of predicting future chatter without delayed observation of chatter evolution. `ChatterNet` satisfies this requirement better than other baselines, because they were all designed for much larger early observation windows. *TiDeH takes 24 hours of observation to outperform* `ChatterNet` *with 1 hour of observation.* Via time-sensitive combination of exogenous and endogenous signals, `ChatterNet` achieves superior performance without network knowledge.

4.2 Ablation of `ChatterNet` Components

We justify the complexity of `ChatterNet`, and show that all its pieces are critical. In Table 3, we present the performances of the

Table 4: (Ablation study) Subreddit-wise MAPE for ChatterNet, ChatterNet-N, and ChatterNet-S all using early observation window of size 1 hour. Subreddits with * are abbreviated names given by: AR→ AskReddit, TD→ The_Donald, WITT→ whatisthisthing, RL→ RocketLegue, TNF→ TheNewsFeed, FF→ Fantasy_Football, NSQ→ NoStupidQuestions, AS→ askscience, WSB→ wallstreetbets, GO→ GlobalOffensive, PF→ personalfinance, UPO→ unpopularopinion, EI→ EcoInternet, AN→ AutoNewspaper, TTF→ TheTwitterFeed, NBB→ newsbotbot, NBTMT→ newsbotTMT, NBMARKET→ newsbotMARKET, BN24→ BreakingNews24hr, PH→ PoliticalHumor, BCAll→ BitcoinAll. Results for some of the subreddits with significant changes due to ablation of exogenous/endogenous signals are highlighted in bold.

Model	**AR***	**TD***	**gaming**	**politics**	**technology**	**Music**	**techsupport**	**WITT***	**news**	**movies**	**RL***
ChatterNet	**26.031**	31.854	24.462	**23.389**	25.686	21.249	25.13	26.002	**29.102**	25.076	24.221
ChatterNet-S	**28.411**	34.712	31.011	**38.105**	36.122	28.310	36.54	27.151	**35.671**	28.510	27.907
ChatterNet-N	**30.008**	35.003	28.949	**25.610**	30.991	27.569	29.171	27.711	**29.342**	26.134	27.886
Model	**Tinder**	**TNF***	**anime**	**india**	**Jokes**	**soccer**	**FF***	**NSQ***	**nfl**	**AS**	**WSB***
ChatterNet	23.861	33.420	26.004	**27.009**	25.562	**24.753**	27.419	25.510	28.911	23.70	23.251
ChatterNet-S	25.945	36.138	28.040	**39.202**	32.787	**33.402**	29.958	29.875	35.007	26.419	24.36
ChatterNet-N	26.020	36.287	29.011	**32.784**	34.095	**30.92**	29.011	31.592	32.019	24.499	24.212
Model	**InNews**	**GO***	**teenagers**	**POLITIC**	**brasil**	**NBA2k**	**bussiness**	**PF***	**nba**	**worldnews**	**UPO***
ChatterNet	29.534	24.993	24.771	27.364	24.77	25.333	26.001	23.122	25.251	24.924	28.037
ChatterNet-S	32.119	27.604	26.759	29.990	26.904	30.424	31.213	31.797	28.751	28.117	30.301
ChatterNet-N	30.54	27.601	27.002	32.013	27.591	28.701	28.10	32.107	26.29	25.023	29.213
Model	**EI***	**AN***	**NBB***	**FIFA**	**BN24***	**BCAll***	**NBTMT***	**TTF***	**PH***	**NBMARKET***	-
ChatterNet	27.001	27.159	24.146	0.754	24.113	28.011	26.112	25.301	24.35	23.109	-
ChatterNet-S	28.994	32.571	32.386	29.778	29.292	27.003	29.203	26.906	24.997	27.904	-
ChatterNet-N	28.923	28.529	25.183	29.022	28.124	27.091	28.114	28.476	26.870	27.878	-

various ablation models described in Section 3.3.5. It is evident that removal of either exogenous or endogenous signals from ChatterNet results in a degraded performance. However, ChatterNet with only endogenous signal slightly outperforms its counterpart with only exogenous signal. This difference does not tell us whether endogenous signals are more important — we need to study their effect for individual subreddits to comment on that.

Removing both signals degrades performance heavily, particularly in the zero-shot setting. This is expected, because ChatterNet-Static in the zero-shot regime is simply a static convolution block mapping submission texts to their future chatter – a regression task juxtaposed with simple text classification engine. The decrease in performance is more evident with the MAPE measure (195.344 and 149.558, respectively for zero-shot and early observation). Even in zero-shot setting ChatterNet-Static utilizes the information of average comment arrival in the subreddit to scale the future chatter accordingly and learn at least a possible ranking of submissions with respect to their future chatter. Additionally we removed this operation as well for ChatterNet-Static; kendall τ for this further ablated model dropped to 0.02.

Comparing the performances of ChatterNet-Static in zero-shot (only submission features) and LSTM-CC (only comment arrival features), one can easily conclude that, when the exogenous and endogenous signals are not taken into account, comment arrival patterns are much powerful indicators of future chatter compared to submission texts.

4.3 Effect of Observation Window and Size of Discussion

Table 5 shows the variation of performance for ChatterNet and LSTM-CC with different sizes of observation window used to aggregate early arrivals of comments. While LSTM-CC shows a steady betterment of performance with increasing observation, ChatterNet takes a quick leap from zero-shot to 15 minute early observation and then reaches a nearly-stationary state. As shown in Figure 4, with longer initial observation, ChatterNet tends to decrease the error rate for predicting high values of chatter.

Table 5: MAPE and kendal-τ scores to predict future chatter using ChatterNet and LSTM-CC, each with varying size of the observation window. *model_name-x* signifies the model uses an observation window x minutes long.

Model	**MAPE**	τ
ChatterNet-0	33.142	0.4042
ChatterNet-15	31.886	0.4278
ChatterNet-30	28.12	0.4302
ChatterNet-45	26.042	0.4361
ChatterNet-60	25.893	0.4439
LSTM-CC-15	196.128	0.0639
LSTM-CC-30	174.667	0.1307
LSTM-CC-45	167.024	0.2259
LSTM-CC-60	152.313	0.3580

Cheng et al. [4] reported increasing uncertainty in predicting larger cascades. We plot the absolute error in prediction vs. ground-truth value in Figure 4, for different early observation windows. We measure the absolute error to predict the size gain after observation. In all four cases, absolute error varies almost linearly with size. However, with longer observation, the slope drops. With a 60 minutes long early observation, absolute error nearly grazes a zero slope line. However, these plots shows the joint effect of increasing observation and decreasing post-observation gold value. Table 7 summarizes how well ChatterNet predicts future chatter with different prediction windows. Again, longer a discussion persists, harder it becomes to predict the final amount of chatter. Also, as can be expected, the zero-shot system tends to suffer more with longer prediction window.

Table 6: Effect of early observation on the chatter prediction performance over different subreddits. We take ChatterNet in zero-shot and early observation setting (ChatterNet+ and ChatterNet++, respectively) and LSTM-CC as models for comparison; results are reported on 10 of the 43 total subreddits – 5 showing most response towards observation (top half), and 5 with least response (bottom half). Abbreviations of subreddits follow the same definitions in Table 4.

Model	AR	Anime	FF	EI	UPO
ChatterNet+	37.184	37.545	39.060	38.219	37.011
ChatterNet++	26.031	26.004	27.419	27.001	28.037
LSTM-CC	110.34	108.69	129.078	121.336	115.414
Model	india	soccer	NBB	business	TNF
ChatterNet+	34.323	31.095	31.866	30.519	33.911
ChatterNet++	27.009	24.753	24.146	26.001	25.686
LSTM-CC	168.91	177.247	163.077	165.325	162.12

Table 7: MAPE and Kendall's τ to predict future chatter using ChatterNet and LSTM-CC, each with varying size of the prediction window. ChatterNet-E_x and ChatterNet-Z_x signify ChatterNet in 1 hour early observation and zero-shot setting using a prediction window x days long, respectively.

Model	MAPE	τ
ChatterNet-E_1	19.020	0.5145
ChatterNet-E_{10}	21.979	0.4610
ChatterNet-E_{20}	25.224	0.4437
ChatterNet-E_{30}	26.042	0.4361
ChatterNet-Z_1	22.198	0.4812
ChatterNet-Z_{10}	25.064	0.4334
ChatterNet-Z_{20}	32.719	0.4098
ChatterNet-Z_{30}	33.142	0.4042

4.4 Subreddit-wise Analysis

Exogenous vs. Endogenous influence. While Table 3 provides useful insights about the roles played by ChatterNet components, drilling down from aggregate performance into different subreddits gives additional insight. As discussed in Section 1, endogenous and exogenous influence manifest themselves differently over different subreddits (which is why we used subreddit embeddings $\mathbf{U}_j$ in both components: influence aggregation and time-evolving convolution). In Table 4, we present the performances of ChatterNet ChatterNet-N, and ChatterNet-S for each of the 43 subreddits.

In some subreddits (e.g., *r/techsupport, r/india, r/business, r/POLITIC, r/InNews*, etc.), ChatterNet suffers more with the ablation of exogenous signal compared to the endogenous one. Most of this subreddits are either directly news related (like *r/news, r/InNews, r/worldnews,* etc.), or very closely governed by what is happening in the real world, like *r/technology, r/business, r/nfl, r/movies,* etc.

Some subreddits are naturally grouped, i.e., they share common topics of discussion, common set of commenting users, etc. Subreddits like *r/Music, r/movies and r/anime* fall into one such group. This sharing of information facilitates ChatterNet-S to perform better compared to ChatterNet-N as it uses the endogenous knowledge in terms of previous submissions, subreddit embeddings, and comment rates. Also there are some particular subreddits (e.g., *r/AskReddit, r/teenagers, r/NoStupidQuestions*) where endogenous information becomes more important than the exogenous one.

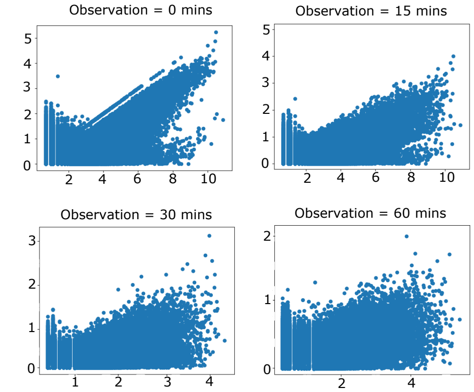

Figure 4: Plots showing absolute error (y-axis) vs. ground truth value (x-axis, log of aggregate comment count to a submission) for ChatterNet designed with different observation windows. With increasing observation window, the volume of discussion remaining to predict (total size − observation size) decreases, corresponding to the shrinking x-axis in the plots. As observation window increases, the error vs. gold value slope decreases, i.e., higher values of chatter become more predictable

Zero-shot vs. Early Observation. Much similar to exogenous and endogenous influence signals, the role of early-observation to predict future chatter differs between subreddits. In Table 6, we explore this phenomena by comparing ChatterNet in zero-shot and early observation settings. We also gauge the performance of LSTM-CC as this ablation variant depends purely on comment arrival in observation window to predict future chatter. For the subreddits in the top half of Table 6 the performance gain with moving from zero-shot to early observation regime is significantly higher compared to those in the bottom half. LSTM-CC follows the same pattern, with MAPE for each of the top-half subreddits being substantially lower than the global average (see Table 2) and higher for the top-half subreddits. Various characteristics of the subreddits might be put as responsible for this: size of the subreddit (large subreddits like AskReddit embody complex dynamics of user interests, resulting in chatter signals that can not be modeled using influence signals alone), small secluded subreddits like EcoInternet with focus of discussion not much available over news or rest of the Reddit, etc.

5 RELATED WORK

Most prior work on engagement prediction requires knowledge of the underlying social network. Such systems are mostly based on modeling information or influence diffusion at a microscopic level [4, 18], recently enhanced with point processes [17, 27]. Separation of exogenous and endogenous influences [5] in the predictive model can increase interpretability and accuracy. Some systems predict if, after observing k instances of influence or transfer along edges, the process will cascade to over $2k$ transfers 'eventually' [4]. Most such systems use a richly designed space of temporal, structural, and contextual features, along with the standard supervised classifiers, to predict the evolving dynamics of the cascades.

In contrast, we do not assume any knowledge of an underlying network. Moreover, we are not tracking the diffusion of any uniquely identifiable content like an image or a hashtag. There is no one-to-one mapping between a specific news story and a network community. In fact, the same event may be reported in multiple news stories. While subredits generally have overlapping interests, the extent of exogenous influence of a news story within a subreddit is topical in nature. Endogenous influence within a community depends on the invisible and possibly transient social links.

Guerini et al. [9] explore the prediction of viral propagation, based solely on the textual features and not the social network structure, which is closer in spirit to ChatterNet compared to network-assisted prediction. They track the spread of specific identified information items (short 'stories'). A story can be submitted only once, unlike multiple submissions on a topic in our setting. They propose hardwired definitions of appreciation, discussion, controversiality, and 'white' and 'black' buzz, then use an SVM classifier to predict such labels successfully. Aswani et al. [2] presented similar studies. Shulman et al. [23] found early adoption a stronger predictor of later popularity than other content features. Weng et al. [25] showed how limited attention of individuals causes competition in the evolution of memes.

Peng et al. [22] seek to predict (as early as possible) emerging discussions about products on social media without information about the social network structure. Like Guerini et al. [9], they engineer a variety of rich features, including author diversity, author engagement, competition from other products, and temporal, content, and user features in a conventional classifier. Their task is thus limited to a vertical domain (products), and the prediction task is discrete classification (will a burst of activity emerge or not). In contrast, we seek to predict quantitative levels of chatter. Unlike both Peng et al. [22] and Guerini et al. [9], we avoid extensive feature engineering and instead focus on the design of a deep network that integrates exogenous and endogenous influences.

Chatter intensity or related quantities have been used for predicting other social outcomes as well. Asur and Huberman [1] used a linear regression to predict their box-office success. Other examples such as election outcome and stock movements are surveyed by Yu and Kak [26]. Thus, our work on chatter intensity prediction opens up avenues toward such compelling downstream applications.

6 CONCLUSION

Activity prediction usually depends on knowledge of the underlying social network structure. However, on several important social platforms, the social network is incomplete, not directly observable, or even transient. We introduce the problem of predicting social chatter level without graph information, and present a new deep architecture, ChatterNet, for this setting. ChatterNet combines deep text representation with a recurrent network that tracks the temporal evolution of the state of a community with latent connectivity. Without knowledge of social network topology, ChatterNet achieves new state-of-the-art accuracy. Here we have regarded chatter as caused by news events but not vice versa, whereas chatter is having increasing effects on the real world. Modeling such feedback effects may be a natural avenue for future work.

REFERENCES

[1] Sitaram Asur and Bernardo A Huberman. 2010. Predicting the future with social media. In *ICWSM*. 492–499.
[2] Reema Aswani, SP Ghrera, Arpan Kumar Kar, and Satish Chandra. 2017. Identifying buzz in social media: a hybrid approach using artificial bee colony and k-nearest neighbors for outlier detection. *SNAM* 7, 1 (2017), 38.
[3] Erik Cambria, Soujanya Poria, Devamanyu Hazarika, and Kenneth Kwok. 2018. SenticNet 5: Discovering conceptual primitives for sentiment analysis by means of context embeddings. In *AAAI*. 1795–1802.
[4] Justin Cheng, Lada Adamic, P Alex Dow, Jon Michael Kleinberg, and Jure Leskovec. 2014. Can cascades be predicted?. In *WWW Conference*. ACM, 925–936.
[5] Abir De, Sourangshu Bhattacharya, and Niloy Ganguly. 2018. Demarcating endogenous and exogenous opinion diffusion process on social networks. In *WWW Conference*. 549–558.
[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.
[7] Subhabrata Dutta, Dipankar Das, and Tanmoy Chakraborty. 2019. Modeling Engagement Dynamics of Online Discussions using Relativistic Gravitational Theory. In *ICDM*. IEEE, 180–189.
[8] Mats Ekström and Adam Shehata. 2018. Social media, porous boundaries, and the development of online political engagement among young citizens. *New Media & Society* 20, 2 (2018), 740–759.
[9] Marco Guerini, Carlo Strapparava, and Gozde Ozbal. 2011. Exploring text virality in social networks. In *ICWSM*. 1–10.
[10] Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A Generic News Crawler and Extractor. In *ICIS*, Maria Gaede, Violeta Trkulja, and Vivien Petra (Eds.). 218–223.
[11] Lala Hu. 2018. Luxury brand communication on social media: A qualitative study of the Chinese market. In *2018 Global Marketing Conference at Tokyo*. 160–161.
[12] Henry Jenkins, Sangita Shresthova, Liana Gamber-Thompson, Neta Kligler-Vilenchik, and Arely Zimmerman. 2018. *By any media necessary: The new youth activism*. Vol. 3. NYU Press.
[13] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188* (2014).
[14] M Laeeq Khan. 2017. Social media engagement: What motivates user participation and consumption on YouTube? *Computers in Human Behavior* 66 (2017), 236–247.
[15] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).
[16] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[17] Ryota Kobayashi and Renaud Lambiotte. 2016. TiDeH: Time-dependent Hawkes process for predicting retweet dynamics. In *Tenth International AAAI Conference on Web and Social Media*.
[18] Andrey Kupavskii, Liudmila Ostroumova, Alexey Umnov, Svyatoslav Usachev, Pavel Serdyukov, Gleb Gusev, and Andrey Kustarev. 2012. Prediction of retweet cascade size over time. In *CIKM*. 2335–2338.
[19] Lian Fen Lee, Amy P Hutton, and Susan Shu. 2015. The role of social media in the capital market: Evidence from consumer product recalls. *Journal of Accounting Research* 53, 2 (2015), 367–404.
[20] Cheng Li, Jiaqi Ma, Xiaoxiao Guo, and Qiaozhu Mei. 2017. DeepCas: An End-to-end Predictor of Information Cascades. In *Proceedings of the 26th International Conference on World Wide Web, WWW 2017, Perth, Australia, April 3-7, 2017*, Rick Barrett, Rick Cummings, Eugene Agichtein, and Evgeniy Gabrilovich (Eds.). ACM, 577–586. https://doi.org/10.1145/3038912.3052643
[21] Ryosuke Nishi, Taro Takaguchi, Keigo Oka, Takanori Maehara, Masashi Toyoda, Ken-ichi Kawarabayashi, and Naoki Masuda. 2016. Reply trees in twitter: data analysis and branching process models. *SNAM* 6, 1 (2016), 1–13.
[22] Sinya Peng, Vincent S Tseng, Che-Wei Liang, and Man-Kwan Shan. 2018. Emerging Product Topics Prediction in Social Media without Social Structure Information. In *WWW Conference (Companion)*. 1661–1668.
[23] Benjamin Shulman, Amit Sharma, and Dan Cosley. 2016. Predictability of popularity: Gaps between prediction and understanding. In *Web and Social Media*. AAAI, 348–357.
[24] Ke Wang, Mohit Bansal, and Jan-Michael Frahm. 2018. Retweet wars: Tweet popularity prediction via dynamic multimodal regression. In *WACV*. 1842–1851.
[25] Lilian Weng, Alessandro Flammini, Alessandro Vespignani, and Fillipo Menczer. 2012. Competition among memes in a world with limited attention. *Sci. Rep.* 2 (2012), 335.
[26] Sheng Yu and Subhash Kak. 2012. A Survey of Prediction Using Social Media. arXiv:1203.1647
[27] Qingyuan Zhao, Murat A Erdogdu, Hera Y He, Anand Rajaraman, and Jure Leskovec. 2015. SEISMIC: A self-exciting point process model for predicting tweet popularity. In *KDD Conference*. 1513–1522.
[28] Zhou Zhao, Lingtao Meng, Jun Xiao, Min Yang, Fei Wu, Deng Cai, Xiaofei He, and Yueting Zhuang. 2018. Attentional Image Retweet Modeling via Multi-Faceted Ranking Network Learning.. In *IJCAI*. 3184–3190.

Deep Exogenous and Endogenous Influence Combination for Social Chatter Intensity Prediction

(Appendix / Supplementary Material)

A CORPUS PREPROCESSING

We use same strategy for text cleaning of both news articles and submissions. After tokenization, replacing URLs, and converting numeric values to their textual counterpart, we set a maximum document frequency of 0.8 (fraction of the total number of news articles and submissions) and minimum document frequency of 5 (absolute count) to exclude stopwords and extremely rare words. We trained the Word2Vec model for 500 iterations with window size set to 10 and output dimension 100. We take the maximum length of texts to be 50 and 100 words for submissions and news articles, respectively.

B DESIGN DETAILS OF `CHATTERNET`

For each of the 43 subreddits represented as one-hot vector, the subreddit embedding layer outputs a 32-dimensional vector. Every weight matrix is randomly initialized using Xavier initialization. All bias matrices are initialized with zeros.

B.1 Organization of Convolution Blocks

`ChatterNet` uses two separate stackings of convolution-maxpool operations: static convolution block and time-evolving convolution block (components 1 and 3 in Figure 3). Internal organizations of the blocks are shown in Figure 5 (static) and Figure 6 (time-evolving). For all the convolution operations in the static block and the branched segments of the time-evolving block we use padding to keep the size of output feature maps to be same as inputs. For the last three convolution operations in the temporal block we do not use any padding (as the kernel size is 1).

All the branches of convolution-maxpooling in both the blocks have number of filters 128, 64, and 32, successively. Last three convolution operations in time-evolving block have filter numbers 64, 32, and 1, successively.

B.2 Parameters of Recurrent Units

The news-aggregating and the submission-aggregating GRUs (see component (2) in Figure 3) both have hidden state size equal to 128. The LSTM layer aggregating comment arrivals in the early observation window uses hidden state of size of 8.

C `CHATTERNET` TRAINING DETAILS

To initialize the word embedding layers, we train a skip-gram Word2Vec model on the training split of the news and submission data. The resulting word vectors are of size 100, and they are further trained while training `ChatterNet`. `ChatterNet` is optimized using Adam optimizer [16], with the learning rate set to 0.00001. As the model works in an online setting, the batch size is set to one. While training `ChatterNet`, we reset the states of stateful GRUs after every epoch (after testing on the validation data).

`ChatterNet` is trained on a Intel Xeon Processor (16 cores, 32 GB RAM) with NVIDIA Quadro K6000 GPU. Each training iteration of full `ChatterNet` takes 13 hours and 22 minutes (roughly). We trained all the models for 25 iterations and save top 5 best models (on validation loss). All the results reported are averaged over these 5 models.

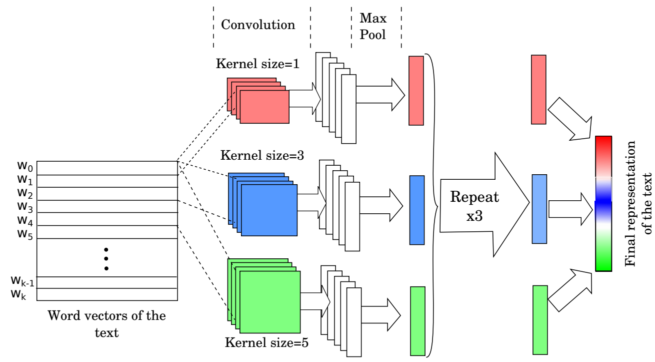

Figure 5: Static convolution of news and submission texts to obtain the latent feature representations. The three parallel branches of convolution-maxpooling is repeated thrice, and then concatenated to produce the feature map. The initial list of the word vectors comes from the word embedding layer.

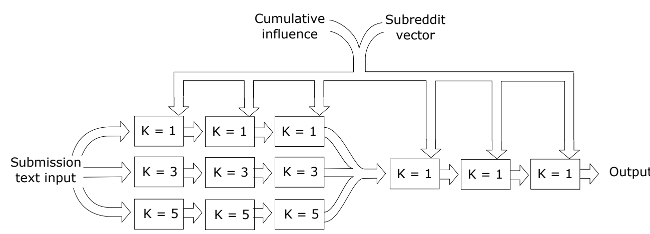

Figure 6: The time-evolving convolution component. K denotes size of convolution filters. Cumulative influence and subreddit vectors corresponding to the input submissions are the control inputs for every convolution layer.