SENTI2POP: Sentiment-Aware Topic Popularity Prediction on Social Media

Jinning Li†, Yirui Gao†, Xiaofeng Gao†*, Yan Shi‡, Guihai Chen†
†Shanghai Key Laboratory of Scalable Computing and Systems, Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai, 200240, China
‡ Department of Computer Science and Software Engineering,
University of Wisconsin Platteville, Platteville, 53818, United States
lijinning@sjtu.edu.cn, gaoyirui98@gmail.com,
gao-xf@cs.sjtu.edu.cn, shiy@uwplatt.edu, gchen@cs.sjtu.edu.cn

***Abstract*—Topic popularity prediction is an important task on social media, which aims at predicting the ongoing trends of topics according to logged historical text-based records. However, only limited existing approaches apply sentiment analysis to facilitate popularity prediction. Public sentiment is worth taking into consideration because the topics with strong sentiment tend to spread faster and broader on social media. In this paper, we propose a novel framework, SENTI2POP, to predict topic popularity utilizing sentiment information. We first adapt a state-of-art popularity quantification method to capture the topic popularity, and then design a novel tree-like network (Tree-Net) combining Long Short-Term Memory (LSTM) and Convolutional Neural Network (CNN) for sentiment analysis. In addition, we propose a sentiment-aware time series prediction approach based on Dynamic Time Warping (DTW) and Autoregressive Integrated Moving Average model (ARIMA) to predict topic popularity. We prove by experiments that SENTI2POP outperforms the existing popularity prediction models on a real-world Twitter dataset by reducing the prediction error. Experimental results also show that SENTI2POP could be applied to improve the accuracy of most non-sentiment popularity prediction models.**

***Index Terms*—Topic Popularity Prediction, Social Media, Sentiment Analysis, Time Series Alignment, Neural Networks**

I. INTRODUCTION

Social media have become an important part of our modern lifestyle. Among the social media platforms, a large amount of text-based information is generated continuously. Predicting the popularity of trending topics is an important part of user behavioral analysis on social media. Accurate popularity prediction can help improve the quality of recommendation systems, online advertising, and information retrieval services.

We observe that the popularity of a topic is strongly associated with the public sentiment towards it. An emotion-intensive topic (no matter positive or negative) would obtain more discussion and attention, thus become more and more popular. Similar observation is also discussed in [1], [2]. Inspired by this, we propose the SENTI2POP model, which computes the sentiment of users and improve the accuracy of popularity prediction with the extracted sentiment information. SENTI2POP (see Fig. 1) includes three procedures, popularity quantification, sentiment analysis, and joint prediction.

We adapt a state-of-art popularity quantification model to define the popularity utilizing text semantic information. Typically, previous works usually use calculations such as the number of re-tweets, likes, and views. However, these simple methods cannot semantically solve the problems of noise and ambiguity. In addition, many social media do not even provide the services of forwarding, likes, or views, which prevent the usage of existing methods. In this paper, we adapt *Term Frequency with Semantic Weight* (TF-SW) [3] to define the textual popularity. This model could capture the semantic relation between words and topics based on the frequency of words weighted by its semantic importance to the topic.

We propose a novel tree-like network for sentiment analysis. The existing sentiment analysis methods usually apply only one of the architecture of Recurrent Neural Network (RNN) [4] and Convolutional Neural Network (CNN) [5]. However, the performances of these RNN-only or CNN-only methods are still limited. Luckily, existing hybrid architectures of long short-term memory (LSTM) and CNN have been proved to be effective in sentence encoding and commonly used in sentiment analysis [6]. Inspired by this idea, we organize these two kinds of networks with a binary tree structure, namely, *Tree-Net*. To better detect the bi-directional information of the input sequence, we apply *Bi-LSTM* [7] to replace LSTM and train the sentiment tendencies of input data, which will be introduced in Section IV. In addition, we also propose an idea to apply the emoticons (emojis) for the labeling of sentiment information.

We propose a joint popularity prediction model combining both popularity and sentiment time series. With quantifying the topic popularity and applying the sentiment analysis, two time series could be obtained including *Popularity Time Series* (PTS) and *Sentiment Time Series* (STS). We prove by experiments that these time series are temporally strongly related, which satisfies the observation that a sentiment-intense topic will become popular more likely. In this paper, we

This work is supported by the National Key R&D Program of China [2018YFB1004703], the National Natural Science Foundation of China [61872238, 61672353], the Shanghai Science and Technology Fund [17510740200], the Huawei Innovation Research Program [HO2018085286], the State Key Laboratory of Air Traffic Management System and Technology [SKLATM20180X], the Tencent Social Ads Rhino-Bird Focused Research Program, and the CCF-Huawei Database System Innovation Research Plan [CCF-Huawei DBIR2019002A].

*Corresponding Author.

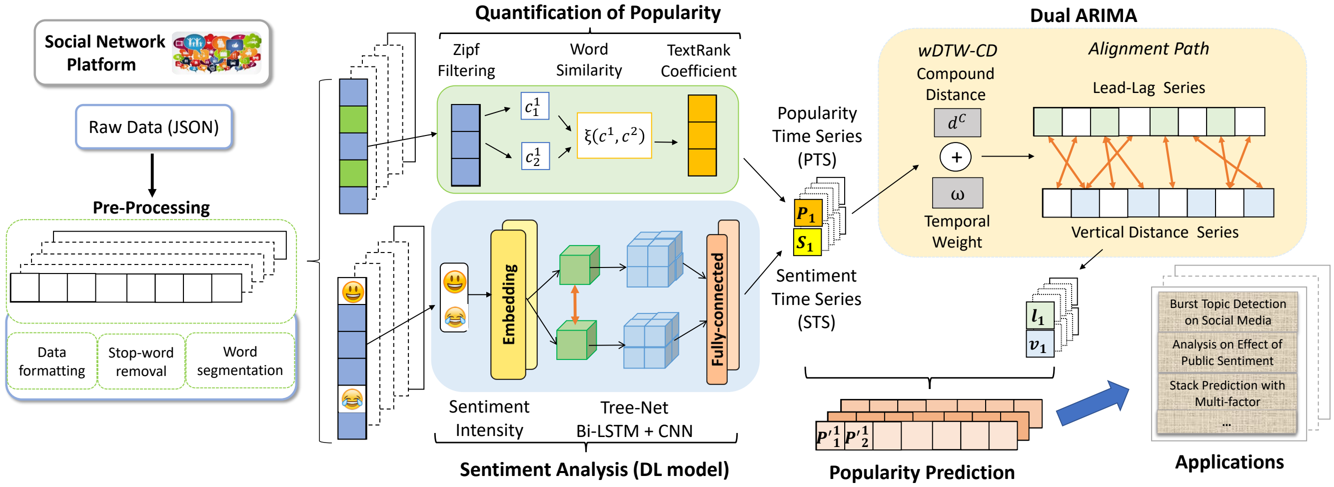

Fig. 1. **Framework of SENTI2POP. The input is text-based data from social media. After the pre-processing procedure, quantification of topic popularity and sentiment analysis are respectively implemented and fed to our topic prediction model. We utilize *TF-WS* in DANCINGLINES as our quantification method. It first uses Zipf's law to obatain contributive words, then calculates the similarities $\xi(\cdot)$ between these words and thus their TextRank coefficients. *TF-SW* generates a popularity time series P_t. For sentiment analysis, the data containing emojis are fed into our *Tree-Net* model and only emojis will be used to calculate the defined sentiment intensity. Then *Tree-Net* can produce a sentiment time series S_t with sentiment intensities. Both P_t and S_t will be combined together into *Dual ARIMA*. wDTW-CD extended from DTW is used to calculate the compound distance and the temporal weight of these time series. Two new series, the lead-lag series l_t and the vertical distance series v_t are produced in this process. Finally, ARIMA maps the total four time series together to predict future popularity. It is worth mentioning that we can not only use this framework for social media topic popularity prediction but also apply it to many other application domains such as burst topic detection or stack prediction.**

propose the *Dual Autoregressive Integrated Moving Average* (Dual ARIMA) for the joint popularity prediction considering the relationship between PTS and STS. Dual ARIMA includes two modules, the *Weighted Dynamic Time Warping with Compound Distance* (wDTW-CD) for temporal alignment of time series and a new design of ARIMA which could jointly predict the popularity with the aligned time series.

We evaluate the performance of SENTI2POP by two comparative experiments. First, we evaluate the performance of sentiment analysis on two public datasets including SST [8] and the CMRD [9]. The experimental result shows that the proposed *Tree-Net* outperforms existing state-of-art sentiment analysis models such as conv-RNN [10] and RNTN [8]. We also verify the effectiveness of our popularity prediction model in the second experiment. The experimental result shows that by taking sentiment into account, the proposed Dual ARIMA outperforms the existing popularity prediction baselines with the minimum prediction error. In addition, we test the improvement of the traditional non-sentiment models after applying the sentiment module proposed in this paper. With the sentiment features extracted by *Tree-Net* and aligned by wDTW-CD, the prediction errors reduce **27.5%** on average than the original model which does not take the sentiment information into account. We prove by experiments that SENTI2POP is less sensitive to the temporal gap between the visible historical period and predicting interval, which means it performs well on the early prediction settings.

II. PROBLEM FORMULATION & PRELIMINARY

SENTI2POP is a predicting framework aiming at predicting future topic popularity by adopting sentiment information based on historical records of text-based data from social media. It receives historical text data of a topic as its input from a given specific social media such as Twitter. In this paper, we define the topic as a series of keywords. We classify one tweet to a topic while it contains the most related keywords of the topic. Practically, we apply the search engine of social media to collect the tweets with manually selected keywords. Data preprocessing mechanism is needed after extracting the original contents from social media. Unrelated contents such as URLs, stopwords, and punctuations are filtered out. We adopt word segmentation to separate words from the sentences. In addition, a kind of Unicode emoticon, *emoji*, is also preserved for sentiment analysis.

Several mathematical expressions for the problem are predefined as below and visualized in Figure 2. (**Time sequence**) We devide time to a interval sequence $T = \langle t^1, t^2, ..., t^n \rangle$ with a constant Δt. (**Representation of words**) We denote w_t^j to represent the j-th word at time t. We use W_t to denote the collection of words occuring at t. (**Representation of records**) We defined a record r_t^i as a subset of W_t at time t, namely $r_t^i = \langle w_t^m, w_t^{m+1}, ...w_t^n \rangle$. The recording set is defined as $R_t = \langle r_t^1, r_t^2, ... \rangle$. (**Contributive words**) There are tons of unrelated noisy words in all tweets relative to a certain topic, we apply Zipf's law as a cut-off threshold mechanism to filter those noisy words and keep the defined contributive word c_t^k, and its collection $C_t = \langle c_t^1, c_t^2, ... \rangle$.

III. POPULARITY QUANTIFICATION

There are various kinds of quantification metrics for topic popularity, most of which only analyze popularity in the level of tweets. However, semantic noise and ambiguity problems will limit the performance of tweet-level metrics. In this section, we are going to introduce a semantic-aware popularity quantification model at the word level, which is more generalized and accurate. We quantify the popularity of a topic

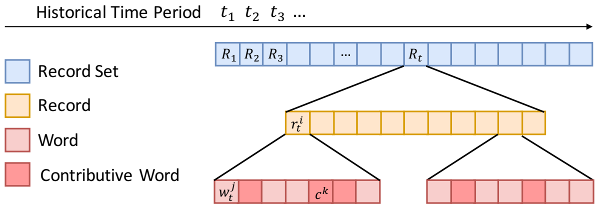

Fig. 2. **Data structure of the concepts. Arrows above represent the historical time period direction when the data are recorded. t_k is the discretized time period. The squares in colors represent different concepts, with the darkest ones the contributive words.**

by combining the frequency and the semantic relationship between the words and the topic together. Similar to the TF-SW metric [3], we define the popularity of a topic as:

$$P_t = \sum_{i=0}^{|C_t|-1} [f(c_t^i) \cdot TR(c^i)], \tag{1}$$

where P_t is the popularity at the time period t. c_t^i is the contributive word in the set of contributive word C_t introduced above. $f(c_t^i)$ denotes the frequency of c_t^i and $TR(c^i)$ denotes the Text Rank coefficient of c^i, a time-independent variable representing the relation between the contributive word c^i and the pre-selected topic. To formalize the $TR(\cdot)$ term in the Eqn. 1, the similarity between word pair $\xi(w_t^i, w_t^j)$ at time period t serves as the basis of our calculation, as is illustrated in DANCINGLINES. We then constructs a weighted undirected graph to represent word similarities by referring to the original PageRank algorithm. In the graph each vertex represents a distinct word and the edge refers to their similarity $\xi(w_t^i, w_t^j)$. With PageRank algorithm, we calculate the importance of each word $TR(w_t^i)$. The formula for TextRank is defined as:

$$TR(w_t^i) = \theta \cdot \sum_{j \to i} TR(w_t^j) \frac{\xi(w_t^i, w_t^j)}{\sum\limits_{k \to j} \xi(w_t^k, w_t^j)} + (1-\theta)\frac{1}{|W|}, \tag{2}$$

where θ represents the probability to continue randomly surfing the edges in the graph. In our work it's set as 0.85 [11].

IV. TREE-NET FOR SENTIMENT ANALYSIS

We introduced how to semantically analyzes the word frequency and textual information of tweets and quantify the Popularity Time Series (PTS). With PTS, many previous time series prediction models could already be applied to predict future popularity. However, the performance of these prediction methods is still limited, because there are many social factors leading to a sharp change of topic popularity. The sudden change of public sentiment has the most significant effect on the change in popularity trends. Luckily, before these topics break out, public sentiment intensity will usually indicate a rapid change. This is an important clue that can be leveraged to predict the popularity more precisely.

SENTI2POP combines the information of public sentiment using a hybrid neural network to optimize the prediction of topic popularity. On most social networks, some records may contain the emoticon emojis. The occurrence of different emojis can be used to evaluate the sentiment intensity of a document, and it is exactly one of the most important and accurate sources related to the sentiment intensity. We defined the sentiment intensity of record r_t^i as $\varepsilon(r_t^i) = \omega_e \cdot \phi_{r_t^i} / |\phi_{r_t^i}|$. ω_e is the manually labeled sentiment vector for the emojis. $\phi_{r_t^i}$ denotes the indicator vector of the record r_t^i. $\phi_{r_t^i}^k = 1$ if the k-th emoji exists in r_t^i, and $\phi_{r_t^i}^k = 0$ otherwise.

To predict the sentiment intensity of any record r_t^i, we propose a tree-like deep neural network, *Tree-Net* in Fig. 3 to learn the mappings accurately. *Tree-Net* is based on a hybrid architecture of Bi-LSTM and CNN. We combine these two networks according to the framework of a binary tree. After training *Tree-Net* with large quantities of tweets, this end-to-end network is capable of figuring out the potential semantic information and mapping it to the corresponding sentiment intensity.

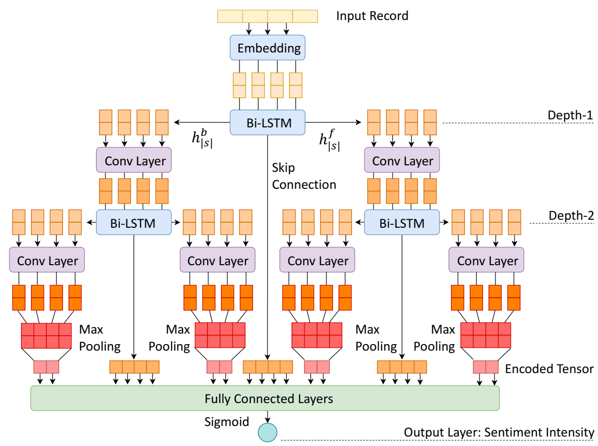

Fig. 3. **The architecture of the *Tree-Net*. *Tree-Net* includes several Bi-LSTM layers, convolutional layers, pooling layers, skip connections and an embedding layer. The input records are sets of words, which are then mapped to a vector space with an embedding layer. A two-depth architecture containing Bi-LSTM layers is followed by convolutional layers. There is a branch at each Bi-LSTM layer, representing the backward and forward information. The convolutional layers after each branching are applied to further encode the information. Skip connections are also applied in every Bi-LSTM layer to avoid the gradient vanishing.**

CNN are widely used to many NLP tasks [12], [13]. The strength of CNN is to process and find the semantic information of local n-grams. However, the CNN architecture is not capable of depicting the overall feature of a series of words. On the contrary, Recurrent architectures such as LSTM are proved to be excellent when processing the sentences with long-range dependency [7], [14], meaning that they are able to learn to forget the useless information of historical words and remember the important ones. Especially, we focus on Bi-LSTM that captures the bidirectional information. A few recent studies use them together to construct sentiment models, but they tend to function in two different parts without being integrated [15].

In our task, since the text data is short, so the analysis of local n-grams becomes quite important. At the same time, the sentences in a record are strongly related in logic. The long term dependency of the sentences, both in the backward and forward direction, should be well addressed. In *Tree-Net*, we

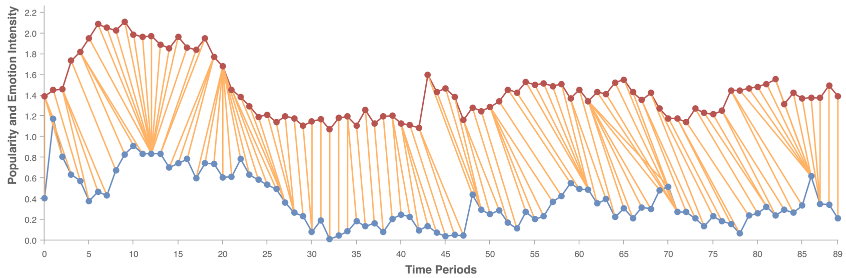

Fig. 4. **Part of the alignment results with wDTW-CD of the topic *Gun Control*. The red line refers to the STS while the blue line being the PTS. We can find that the overall trend of the aligning lines inclines to the STS, which means the emotion change leads to the trend of popularity in this case.**

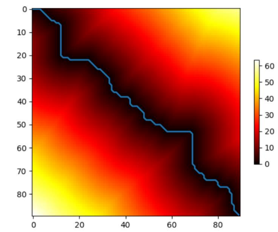

Fig. 5. **The heat map of wDTW-CD. The blue line represents the alignment path Z.**

combine both CNN and Bi-LSTM into a single model with the aim of centralizing the two respective advantages. Besides, to train a deep and complex neural network, skip-connection is a useful trick to improve the performance of neural networks and help them combine both of the original and deep-encoded information. With *Tree-Net*, we could evaluate the sentiment intensity of all the record $\varepsilon(r_t^i)$ at time t. Then we could induce the sentiment intensity S_t of the topic at time period t, $S_t = \frac{1}{|R_t|}\sum_i^{|R_t|}\varepsilon(r_t^i)$.

V. Dual ARIMA for Popularity Prediction

ARIMA is one of the most classical methods used to predict future time series [16] as a simple and quick model. However, the performance of the original ARIMA is limited when the distribution of the time series is non-stationary. In order to make use of ARIMA's advantage and meanwhile solve this problem, we adapt the wDTW-CD algorithm to implement a temporal matching between PTS and STS. wDTW-CD produces an extended temporal lead-lag series l' and a vertical distance series v' from the original two series. Then, instead of using the information of the original time series, *Dual ARIMA* estimates the popularity in the future according to these two series, which possess the stationarity. At the same time, *Dual ARIMA* automatically combines the result of a single ARIMA to achieve greater robustness.

wDTW-CD Algorithm *Dynamic Time Warpping* (DTW) algorithm could find an optimal global alignment between two time series and exploiting temporal distortions between them. However, since only the local minimum cost is considered but not the global optimum, classic DTW with Euclidean distance may suffer from *far-matching* and *singularity* problems when aligning PTS and STS. The far-matching problem illustrates that the alignment of PTS and STSs data points might be temporally too far away, while the singularity problem indicates that a single point is likely to be aligned to too many points of another time series. A more robust and reasonable adaptation should be added into the original DTW algorithm.

To solve the far-matching and singularity problems of classic DTW, we introduce the *compound distance* and *temporal weight* concepts based on DTW algorithm and propose the adaptive DTW algorithm, wDTW-CD. The compound distance is defined as $d_{i,j}^C = \sqrt{d_{i,j}^E \cdot d_{i,j}^D}$. $d_{i,j}^E$ is the Euclidean distance. $d_{i,j}^D = |D(P_i) - D(S_j)|$ is the derivative distance, where $D(P_i)$ and $D(S_j)$ are the estimated derivatives at the points P_i and S_j of PTS and STS respectively. Temporal weight is defined as $\omega_{i,j} = \frac{1}{1+e^{-\eta(|i-j|-m)}}$, where η decides the overall penalty level, which can be adjusted for different patterns of PTS and STS. Factor m is called *prior estimated time difference*, it functions in tracing and controlling the current temporal distance. If the temporal distance between P_i and S_j is larger than m, the alignment will obtain a relatively high penalty. The final objective of dynamic time warpping process is then given by a summation of both the compound distance and the temporal weight for the two time series $d_{i,j} = d_{i,j}^C + \omega_{i,j}$. After applying the wDTW-CD algorithm, the alignment path Z can be generated, which is visualized with heat map shown in Fig. 5. Z contains pairs of time, for example, (t_m, t_n) means t_m in the first series and t_n in the second series are aligned. The aligned PTS and STS are visualized in Fig. 4, providing an intuition on one certain topic's popularity and sentiment.

Prediction with Dual ARIMA Applying Deniel test, we found that PTS and STS are always not or only weakly stationary. It is not suitable to use the original ARIMA to predict the time series directly. After utilizing the novel wDTW-CD, we propose the *Dual ARIMA* model which takes an indirect approach. It uses the temporal lead-lag series l and vertical distance series v induced by the wDTW-CD as the target series. These two series are proved to be stationary with the Deniel test. Therefore, we make use of the extended l' and v' to estimate and predict the popularity.

We define the *temporal lead-lag series* l using the wDTW-CD path Z as $l_t = z_t^1 - z_t^0$. Similarly, the *vertical distance series* v is defined as $v_t = P(z_t^1) - P(z_t^0)$, With both the two target series defined, we then apply ARIMA to predict and extend l and v.

The ARIMA forecasting equation for a stationary time series is a regression-type equation in which the predictors consist of the lags of the dependent variable and lags of the forecast errors. For a non-seasonal ARIMA(p,d,q) model, the projected value is computed by $(x_t - \mu) - \phi_1(x_{t-1} - \mu) - \cdots - \phi(x_{t-p} - \mu) = \epsilon_t - \theta_1\epsilon_{t-1} - \cdots - \theta_q\epsilon_{t-q}$, where p represents the number of AR (Auto-Regressive) terms and q denotes the number of MA (Moving Average) terms. Here we use the *Akaike information criterion* (AIC) law to measure the value of p and q. To predict the series of P, l and v, we first use the single ARIMA to predict the values P', l' and v'.

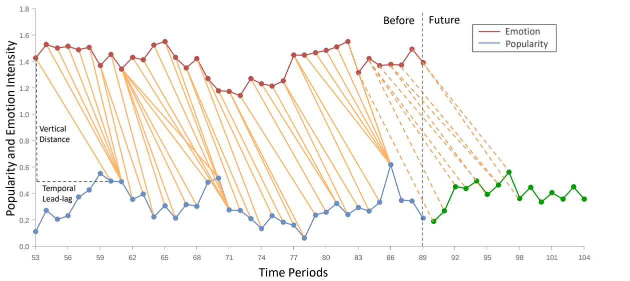

Fig. 6. **The prediction method of *Dual ARIMA*. The extended vertical distance series and temporal lead-lag series applied to predict the green points, by linking the dotted line.**

Then, we apply them on the *Dual ARIMA* model and further calculate the results:

$$\min \ AIC = n \ln \hat{\sigma}_{\epsilon}^2 + 2(p + q + 2), \tag{3}$$

$$P(t) = \alpha \left[S(t + l'(t)) - v'(t) \right] + (1 - \alpha) P'(t), \tag{4}$$

where $P(t)$ is the topic popularity and $S(t)$ is the sentiment intensity defined as above. To make the terms more meaningful, we define $\alpha = 1 - \frac{2}{\pi}\beta$ and $\beta = \max(-\arctan \frac{v'(t)}{l'(t)}, 0)$. In this way, $P(t)$ is calculated using the results by the single ARIMA together with the two stationary series we defined above. Visualization of this prediction is shown in Fig. 6.

VI. Experiments

A. Performance of Sentiment Analysis

Tree-Net is proposed in Section IV to analyze sentiment information of short texts. To evaluate the performance of *Tree-Net*, we operate comparative experiments on two sentiment classification datasets, including SST [8] and the CMRD [9]. SST contains a totally $239,232$ sentences and phrases. In CMRD dataset, $5,000$ subjective and $5,000$ objective processed sentences are included.

*1) **Comparative Methods**:* We select five representative baselines to compare with our *Tree-Net* model, including *convolutional neural network* (CNN) [17], *Fully-Connected Layers* (FC) [18], recursive neural networks *Matrix-vector Recurrent Neural Network* (MV-RNN) [19], *Recursive Neural Tensor Network* (RNTN) [8], and *convolutional-RNN* [10].

*2) **Results and Discussion**:* We use the above baselines to detect the sentiment intensity as what *Tree-Net* does. We report the prediction accuracy in Table I.

TABLE I
Result of Sentiment Intensity Detection

Model	SST / %	CMRD / %
CNN	86.73	92.16
FC	86.24	90.79
MV-RNN	82.9	-
RNTN	85.4	-
conv-RNN	87.71	94.13
Tree-Net	**88.13**	**94.77**

From Table I, we can find that the compound architecture works in the best manner. Our *Tree-Net* achieves the best performance in both of the two datasets. In the Stanford Sentiment Treebank dataset, *Tree-Net* gets the largest accuracy of **88.13%**, while in the Cornell Movie Review Dataset, it is still the most outstanding model with an accuracy of **94.77%**. Notice that the other compound model conv-RNN stands out among all the baselines. Besides, we can see that the deep neural networks like CNN and fully-connected neural network perform better than the structural recursive neural networks on these two datasets.

B. Performance of Topic Popularity Prediction

*1) **Datasets**:* We evaluate the prediction performance on a real-world Twitter dataset collected with Twitter API tools. We first manually assign several keywords for a specific topic. Then, we use the keyword filtering method provided by the API to collect the related tweets of a specific topic. In total, we collected 1.6×10^9 tweets for 3 months (*from 12/23/2017 to 3/18/2018*) on 8 topics, including *Gun Control, Trump, Immigration, AI, HIV, IOS11, Wildlife*, and *Air Quality*. Among these 1.6×10^9 tweets, only about 1.88×10^7 tweets contain the emoji emoticons. The tweets in the last week (from 3/12/2018 to 3/18/2018) are used as the test set containing about 72 million tweets while the left tweets are used for training. We train the Tree-Net with all the tweets containing emojis and inference the sentiment intensity for all tweets.

*2) **Comparative Methods**:* We focus on four popular time series prediction models on stage including the *Gated Recurrent Unit* (GRU) [20], *convolutional neural network* [21], *Multi-layer Perceptron* (MLP) and the traditional ARIMA.

*3) **Results and Discussion**:* In this experiment, we set the historical time period visible to the prediction model as 3 days (72 intervals). The objective of prediction models is to predict the popularity at the next following interval. We train 20 epochs for all the trainable deep learning methods. RMSE is used as our evaluation metric for popularity prediction. The resulting RMSE errors between ground truth and predicted popularity are shown in Table II.

Table II shows the prediction RMSE of different models with and without sentiment information on 8 topics. In Table II, the improvement rate in percentage before and after applying sentiment information is also reported. In the case of popularity without sentiment, the effects of prediction given by the GRU, CNN, MLP, and ARIMA models are similar. However, after combining sentiment information extracted by *Tree-Net*, we observe a consistent improvement of performances in all models, indicating by considering sentiment factors, the performance of the model is getting better.

Our sentiment *Dual ARIMA* model achieves the best performance compared to other models in **6** out of 8 topics, involving 4 pairs of models with and without sentiment. At the same time, for the improvement rate in percentage, it has the biggest improvement in **5** out of 8 topics.

C. Performance of Early Prediction

We have introduced the experimental result predicting the popularity of the next following time interval with a visible historical time period of 3 days. One interesting question is how the performance of prediction models varies when there is a temporal gap between the historical time period and the

TABLE II
RMSE RESULTS OF THE PREDICTION MODELS.

Topic	**Gun Control**				**Trump**				**Immigration**				**AI**			
Model	CNN	GRU	MLP	ARIMA	CNN	GRU	MLP	ARIMA	CNN	GRU	MLP	ARIMA	CNN	GRU	MLP	ARIMA
Non Senti.	0.37	0.33	0.29	0.30	0.35	0.33	0.32	0.32	0.36	0.34	0.33	0.33	0.37	0.36	0.30	0.34
Sentiment	0.31	**0.24**	0.28	0.25	0.29	0.26	0.26	**0.22**	0.3	**0.24**	0.25	0.25	0.32	0.28	0.27	**0.23**
Improv.(%)	16.2	**27.3**	3.5	16.7	17.1	21.2	18.8	**31.3**	16.7	**29.4**	24.2	24.2	13.5	22.2	10.0	**32.4**

Topic	**HIV**				**iOS11**				**Wildlife**				**Air Quality**			
Model	CNN	GRU	MLP	ARIMA	CNN	GRU	MLP	ARIMA	CNN	GRU	MLP	ARIMA	CNN	GRU	MLP	ARIMA
Non Senti-	0.36	0.30	0.32	0.35	0.36	0.34	0.32	0.32	0.39	0.36	0.30	0.31	0.37	0.30	0.31	0.33
Senti-	0.34	0.26	**0.23**	**0.23**	0.29	0.26	0.26	**0.24**	0.34	0.27	0.27	**0.24**	0.29	**0.24**	0.26	0.22
Improv.(%)	5.6	13.3	28.1	**34.3**	19.4	23.5	18.8	**25.0**	12.8	**25.0**	10.0	22.6	21.6	20.0	16.1	**33.3**

predicting time interval. The length of the gap determines how early the prediction is.

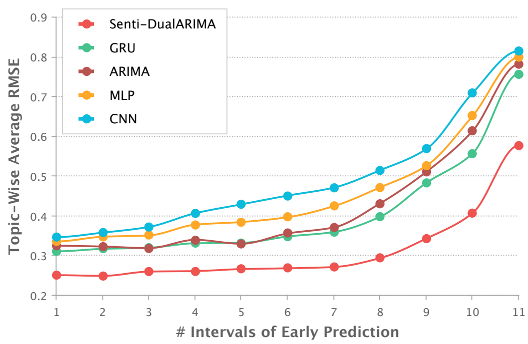

Fig. 7. **The RMSE under different numbers of early prediction intervals (temporal gap). The vertical axis represents the average RMSE on the** 8 **topics, the lower the better.**

The result of RMSE under different temporal gap is shown in Figure 7. We could observe that with the temporal gap increases, the RMSE of all the models increases as well. This result is reasonable since the early prediction of popularity is a harder process than single-step prediction. From the figure, we could also find that the proposed Senti-DualARIMA model is less sensitive to the temporal gap compared with other baseline models including the original ARIMA algorithm. We propose that the joint prediction with sentiment information could also improve the performance on early popularity prediction.

VII. CONCLUSION

In this paper, we designed SENTI2POP, a novel framework to predict topic popularity on social media considering sentiment intensity. Three major components in this framework were discussed in detail: *TF-SW*, a novel topic popularity metric raised recently; *Tree-Net*, a hybrid sentiment model combining Bi-LSTM and CNN, and *Dual-ARIMA*, our topic popularity prediction model utilizing wDTW-CD algorithm. Experiments confirm that sentiment intensity analysis benefits the accuracy of topic popularity prediction. Our SENTI2POP framework outperforms existing popularity prediction models. Possible future directions include applying this framework to other domains such as burst topic detection and stack prediction with sentiment factor considered.

REFERENCES

[1] S. Stieglitz and L. Dang-Xuan, “Emotions and information diffusion in social mediasentiment of microblogs and sharing behavior,” *Journal of management information systems*, vol. 29, no. 4, pp. 217–248, 2013.
[2] E. Ferrara and Z. Yang, “Quantifying the effect of sentiment on information diffusion in social media,” *PeerJ Computer Science*, vol. 1, p. e26, 2015.
[3] T. Gao, W. Bao, J. Li, X. Gao, B. Kong, Y. Tang, G. Chen, and X. Li, “Dancinglines: an analytical scheme to depict cross-platform event popularity,” in *DEXA*, 2018, pp. 283–299.
[4] F. Figueiredo, J. M. Almeida, M. A. Gonçalves, and F. Benevenuto, “Trendlearner: Early prediction of popularity trends of user generated content,” *Information Sciences*, vol. 349, pp. 172–187, 2016.
[5] L. Hong, O. Dan, and B. D. Davison, “Predicting popular messages in twitter,” in *WWW*, 2011, pp. 57–58.
[6] M. Abdullah, M. Hadzikadicy, and S. Shaikhz, “Sedat: Sentiment and emotion detection in arabic text using cnn-lstm deep learning,” in *ICMLA*, 2018, pp. 835–840.
[7] A. A. Sharfuddin, M. N. Tihami, and M. S. Islam, “A deep recurrent neural network with bilstm model for sentiment classification,” in *ICBSLP*, 2018, pp. 1–4.
[8] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *EMNLP*, 2013, pp. 1631–1642.
[9] B. Pang and L. Lee, “A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts,” in *ACL*, 2004.
[10] C. Wang, F. Jiang, and H. Yang, “A hybrid framework for text modeling with convolutional rnn,” in *SIGKDD*, 2017, pp. 2061–2069.
[11] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text,” in *EMNLP*, 2004.
[12] Y. Kim, “Convolutional neural networks for sentence classification,” *arXiv preprint arXiv:1408.5882*, 2014.
[13] D. Zeng, K. Liu, Y. Chen, and J. Zhao, “Distant supervision for relation extraction via piecewise convolutional neural networks,” in *EMNLP*, 2015, pp. 1753–1762.
[14] C. Zhou, C. Sun, Z. Liu, and F. Lau, “A c-lstm neural network for text classification,” *arXiv preprint arXiv:1511.08630*, 2015.
[15] S. Minaee, E. Azimi, and A. Abdolrashidi, “Deep-sentiment: Sentiment analysis using ensemble of cnn and bi-lstm models,” *arXiv preprint arXiv:1904.04206*, 2019.
[16] A. A. Ariyo, A. O. Adewumi, and C. K. Ayo, “Stock price prediction using the arima model,” in *ICCMS*, 2014, pp. 106–112.
[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
[18] N. Limsopatham and N. Collier, “Modelling the combination of generic and target domain embeddings in a convolutional neural network for sentence classification,” in *ACL*, 2016.
[19] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, “Semantic compositionality through recursive matrix-vector spaces,” in *EMNLP*, 2012, pp. 1201–1211.
[20] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.
[21] Y. Zhang and A. Jatowt, “Image tweet popularity prediction with convolutional neural network,” in *ECIR*, 2019, pp. 803–809.