Attend2trend: Attention Model for Real-Time Detecting and Forecasting of Trending Topics

Ahmed Saleh
Kiel University
ZBW - Leibniz Information Centre for Economics
Kiel, Germany
a.saleh@zbw.eu

Ansgar Scherp
University of Stirling
Scotland, United Kingdom
ansgar.scherp@stir.ac.uk

***Abstract*—Knowing what is increasing in popularity is important to researchers, news organizations, auditors, government entities and more. In particular, knowledge of trending topics provides us with information about what people are attracted to and what they think is noteworthy. Yet detecting trending topics from a set of texts is a difficult task, requiring detectors to learn trending patterns while simultaneously making predictions. In this paper, we propose a deep learning model architecture for the challenging task of trend detection and forecasting. The model architecture aims to learn and attend to the trending values' patterns. Our preliminary results show that our model detects the trending topics with a high accuracy.**

I. Introduction

In the last decades, modeling of time series has received a lot of attention due to the growing need to have tools that facilitate decision making [1]. Time series can exhibit not only regular patterns such as trends cycles, and seasonality, but also irregular patterns like structural changes, atypical sequences, effects of calendar days, etc [6]. The relationship between these patterns and time series are considered as an unknown independent variable. For this reason, the problem of detecting and forecasting time series values is challenging.

Various studies attempted to detect the anomalous behavior of time series [5], and analyze the popularity of certain topics. In the latter case, Cheng et. al. showed that national Twitter trends reflect the interests of the inhabitants [3]. On the basis of this fact, Althoff et al. presented a novel approach to analyze and forecast the life cycle of trending topics [2]. They used three different datasets that are sampled from Wikipedia, Twitter, and Google news. Although the work of Althoff et al. showed that the trends' life cycle of semantically related topics is very similar, the authors' sequence matching and forecasting approach does not benefit from these similar patterns.

Motivated by this, we introduce a trend detection and forecasting approach, based on attention neural networks, which is able to learn the trending patterns of time series as well as forecasting the upcoming values. Our approach is composed of three main components: (1) The input text stream, e.g. Tweets, is passed to a sequence tagging model that extracts the main entities from the text associated with their frequency time series. (2) The frequency time series of these entities is transformed into volatility time series which measures the dispersion around the mean. (3) We utilize our forecasting model to learn the volatility time series, with a special attention units to memorize the patterns of the peaking values, and forecast trends. Please note that we used the state-of-art model for sequences tagging as a black-box, since the main contribution of the paper is the trend detection model.

We expect that the proposed trend detection method is applicable to any time series, despite utilizing only Twitter time series as a motivating example. For this purpose, we intend to apply our approach to other datasets such as Wikipedia page view statistics[1].

The remainder of the paper is organized as follows: Our proposed trend detection and forecasting approach is described in Section II. We introduce the evaluation procedures and results in Section III and the work in progress in Section IV.

II. Proposed model: Attend2trend

We design a preliminary attention model that is capable of detecting the trending topics in text streams. Attention models are encoder-decoder models that support the decoder to pick only the encoded inputs that are important for each step of the decoding process [10].

Assume that we have a sequence of n input time steps $Input = \{inp_1, ..., inp_{n-2}, inp_{n-1}, inp_n\}$ that contains an eBay Atlas trend [5] on inp_n, our model is required to forecast the trend as early as possible i.e. on inp_{n-3}, inp_{n-2} and detect the trending value inp_n. In the ebay Atlas trend, the trend occurs when one value of the time series is equal to or greater than 3σ of the time series values. We define an output vector of two sets. The first set consists of the emerging trend values, while the second set represents the next time step of the time series that contains the ebay Atlas trend itself. For example, in the following time series $S = \{1, 3, 2, 6, 7, \mathbf{11}, \mathbf{13}\}$, the expected input vector is $Input = \{1, 3, 2, 6, 7, \mathbf{11}\}$ while the output sequence vector will be $Output = \{\{0, 0, 0, 6, 7, \mathbf{11}\}, \{\mathbf{13}\}\}$. In this case, we teach the network to attend to the trending values pattern (in bold), and forecast whether the trend will continue.

The model is composed of three main layers; namely encoder, attention and decoder. In the first layer, the encoder

The final authenticated version is available at https://doi.org/10.1109/ICDMW.2018.00222

[1] $https\ //en.wikipedia.org/wiki/Wikipedia : Pageview\_statistics$

encodes the entire input sequence, for each time step, into a fixed length vector [4]. The attention layer calculates the importance of the encoded vector e at decoding step j. Therefore, the attention model requires access to the output from the encoder for each input time step, and the internal hidden state of the decoder cell, h_{i-1}.

In order to calculate the attention probabilities $\alpha = (\alpha_0, , \alpha_n)$, a feed-forward neural network calculates the formalized importance of the encoded vector in step i in predicting j (Equation 1).

$$\beta_{ij} = V \cdot \tanh(W_1 \cdot e_i + W_2 \cdot h_j) \tag{1}$$

Then, the softmax operation normalizes the probabilities of the encoded vectors by multiplying each encoded vector by its weight, in order to obtain a time dependent input encoding which is fed to each step of the decoder.

$$\alpha_{ij} = \frac{exp(\beta_i)}{\sum_{k=0}^{n} exp(\beta_i)}, c_j = \sum_{k=0}^{n} \alpha_{k,j} e_k \tag{2}$$

Afterwards, the model calculates the context vector. This vector summarizes the importance of the different encoded values (Equation 2). The decoder steps through the output time series while reading from the context vector. Therefore, the attention model computes a weighted representation of the whole input sequence for each slide of the decoder, and the decoder can learn to pay attention to the trending values.

III. Evaluation and Results

From the sequence labeling approach, we obtain a time series of entities frequencies. This time series is considered as the input layer for our proposed attention model. For each entity time series, the model detects the trending values. To evaluate the time series prediction performance, we use Mean Squared Error (MSE). MSE measures the average squared difference between the real trending values and what is predicted.

Datasets: We use the *Tweets2011 dataset*. The dataset is provided by the National Institute of Standards and Technology (NIST). It was published as a part of the micro-blog track in 2011. The dataset contains Twitter identifiers of approximately 16 million tweets. The tweets identifiers cover the period between January 23rd and February 8th, 2011.

Entities Extraction: In order to extract entities from a set of texts, a dataset to train the named-entity recognition (NER) model is required. Such a dataset usually consists of texts and labels for each token of the text. We train the model with the W-NUT dataset [9]. The dataset consists of 2,400 tweets and 34,000 entities with 10 different annotated types. We use the model that obtains the state-of-art performance in NER. The model uses a bi-LSTM and CRF with character embedding [8]. We use cross validation to induce the optimal model training parameters. The trained model was then used to extract all entities from the Tweets2011 collection. From Tweets2011, we extracted $331,913$ entities.

Detecting Trends: After generating the frequency time series of entities, we use the min-max normalization to normalize the series values. Afterwards, we pass the normalized time series values for each entity to our trend detection model, attend2trend. As with every other deep learning model, the training process is computationally expensive due to the high number of parameters that should be tuned. In order to find the best set of parameters that reduces the mean squared testing error, our model has been executed with different parameters, each execution consists of 150 iterations (epocs). The initial set of parameters to choose from are: (1) $Windowsize := [0, 150]$, (2) $Activationfunction := \{Tanh, Sigmoid\}$, (3) $Batchsize := [1, 10]$, (4) $Numberofhiddenunits := [100, 200]$. We use the adaptive gradient descent algorithm Adam for tuning the learning rates [7].

After finding the best parameters for each dataset, we apply the holdout cross validation to train and test the model. From Tweets2011 dataset, only 280 entities (out of $331,913$) contain obvious eBay Atlas trends. The preliminary results shows that our encoder-decoder model, powered by the attention units, was able to detect the trends with an accuracy of 94.05% and RMSE of 1.98, comparable with the accuracy of 76.19% and RMSE of 7.80 of the classic LSTM encoder-decoder model.

IV. Work in Progress

Our ongoing work consists of the following components: (1) calculating not only the accuracy of detecting trends, but also forecasting trends. (2) comparing our model with other baseline forecasting models that use a short history window of the time series itself i.e. a naive forecast (tomorrow's behavior is the same as today's), linear, median, and average trends. (3) utilizing other datasets such as Wikipedia page view statistics for testing our model.

Acknowledgement This work was supported by the EU under grant agreement H2020-693092 MOVING.

References

[1] R. Adhikari and R. Agrawal, "An introductory study on time series modeling and forecasting," *arXiv preprint arXiv:1302.6613*, 2013.

[2] T. Althoff, D. Borth, J. Hees, and A. Dengel, "Analysis and forecasting of trending topics in online media streams," in *Proceedings of the 21st ACM international conference on Multimedia*. ACM, 2013, pp. 907–916.

[3] R.-C. Chen, D. Spina, W. B. Croft, M. Sanderson, and F. Scholer, "Harnessing semantics for answer sentence retrieval," in *Proceedings of the Eighth Workshop on Exploiting Semantic Annotations in Information Retrieval*. ACM, 2015, pp. 21–27.

[4] 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," 2014.

[5] D. Goldberg and Y. Shan, "The importance of features for statistical anomaly detection." in *HotCloud*, 2015.

[6] A. C. Harvey, *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

[7] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[8] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," *arXiv preprint arXiv:1603.01360*, 2016.

[9] B. Strauss, B. Toma, A. Ritter, M.-C. de Marneffe, and W. Xu, "Results of the wnut16 named entity recognition shared task," in *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, 2016, pp. 138–144.

[10] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International Conference on Machine Learning*, 2015, pp. 2048–2057.