ISSN No:-2456-2165

Self-Attention GRU Networks for Fake Job Classification

Ankit Kumar
University of Delhi
New Delhi, India

Abstract:- This paper analyses the Employment Scam Aegean Dataset and compares various machine learning algorithms including Logistic Regression, Decision Tree, Random Forest, XGBoost, K-Nearest Neighbor, Naïve Bayes and Support Vector Classifier on the task of fake job classification. The paper also proposes two self-attention enhanced Gated Recurrent Unit networks, one with vanilla RNN architecture and other with Bidirectional architecture, for classifying the fake job from real ones. The proposed framework uses Gated Recurrent Units with multi-head self-attention mechanism to enhance the long term retention within the network. In comparison to the other algorithms, the two GRU models proposed in this paper are able to obtain better result.

Keywords:- *Fake Job Classification; Text Classification; Gated Recurrent Unit; Recurrent Neural Networks.*

I. INTRODUCTION

21st century world is the world of data. There has never been more data available to humans at once than now. Data is available in various formats – texts, audios, videos, images, graphs and more. There was a time when reaching people or accessing things was not easy, but with the advent of internet everything has changed. People are one text or internet call (audio or video) away from each other irrespective of their geographical locations. Books, journals, news, recruitments-information regarding anything and everything was difficult to access earlier, again with internet, it has become easier to access data or such information. Within three decades of arrival of internet, we have moved from a time of not enough data to way too much data. With so much data available at once, we are at advantage. However, just as there is some bane associated with every boon, this availability of too much data also has some hidden issues. Especially when there is no validity of the data. With the advent of social media platforms it has become really easy to share information obtained from these data with people. However, this ease has brought a major issue with it. People can and do share information with other people without verifying it. An information that is not verified could pose some real threat to people using that data. For instance, a famous journalist in India thought she got a job to teach at one of the top ranked university in the world. She quit her job to accept this teaching position. However, later she got to know that the job offer that she received was fake and there was no teaching job for her. She had left her journalist job by then. This is just one such instance of people falling in the trap of fake or unverified information.

A large amount of data that we encounter is text based. Text data requires considering semantic as well as syntactic significance of words. With deep learning, Natural Language Processing (NLP) has accomplished great heights. It has empowered our machines to examine, comprehend and choose important contexts out of the compositions. Nowadays, Recurrent Neural Network (RNN) has come up as an empowering alternative to withstand the test of time not just on one but numerous text-based jobs.

Recurrent Neural Networks have been utilized for different applications like text classification [1, 2, 3, 4], speech recognition [5], language translation [6], image captioning [7], and various others. Speculatively, vanilla Recurrent Neural Networks show energetic common conduct for a time series task. However, Hochreiter [8] and Bengio et al., [9] proved that vanilla Recurrent Neural Networks are frail to dispersing or detonating slopes. To overcome this issue of frailing slope, Hoschreiter proposed Long Short-Term Memory (LSTM) in his 1997 paper [10]. LSTM is a combination of three gates namely input, forgets and output gates. The three gates together solve the issue of the slope. A more summarized adaptation of LSTM called Gated Recurrent Unit (GRU) was proposed in 2011 by Cho et al., [11]. Both the LSTM and GRU have been used in RNN architecture for various tasks and have resulted in many state-of-the-art results. Since GRU has only two gates instead of three as is the case with LSTM, GRUs are computationally faster than LSTMs.

The rest of the sections of this paper are structured as follows: Section 2 details about GRU cell and the use of GRU based RNN architectures for text classification. Besides this the section details about the calculation of self-attention weights. In section 3, we have given the details our models. Section 4 includes the details of datasets, implementations, results and the various observations that we have made based on the outcomes of our experiments. We conclude this paper in section 5.

II. BACKGROUND

A. Recurrent Neural Networks for Text Classification

Recurrent neural network is a sequential network in which output at each step is calculated as the function of its current input and the outputs obtained from the previous inputs. With the recent progression within the field of text classification utilizing RNNs, recurrent networks are being utilized for an assortment of errands. Irsoy et al., [12] in 2014, used RNN for opinion mining. Pollastri et al., [13], in 2002, used RNNs for estimating the protein secondary structure. Tang et. al., [14] did sentiment classification using the gated recurrent network in 2015. Arevian [15], in 2007, used RNN to classify real-life text data. Melsin et al., [17], in 2015, used RNNs for the task of slot filling. A combination of RNN and Convolution Neural Network (CNN) was used by Lai et al., [16] in 2015 to classify texts. Liu et al., [2] used recurrent neural networks for implementing a joint intent detection and slot filling model in 2016. Lee et al., [18] also used the RNNs in combination with CNNs to classify short texts in 2016. In text classification RNNs are being employed for various tasks. Therefore, it seems natural to employ RNNs for sequence based tasks.

B. Gated Recurrent Unit

In GRU, as depicted in Fig 1, activation h_t^k for the k^{th} recurrent unit at time *t* is calculated as the linear interpolation between the previous activation h_{t-1}^k and future candidate activation $\bar{h}_t^k$. It is given as

$$h_t^k = (1 - z_t^r)h_{t-1}^r + z_t^r \bar{h}_t^k \qquad (1)$$

The update gate z_t^k decides what information will be updated by the unit. It is computed as

$$z_t^k = \sigma(W_z x_t + U_z h_{t-1})^k \qquad (2)$$

Reset gate allows GRU to reads the new input as if it is the first word of the sequence by forgetting the previous computations done on the input and. It is calculated as:

$$r_t^k = \sigma(W_x x_t + U_r h_{t-1})^k \qquad (3)$$

Here W_r and W_z are the weights from input to hidden layer. σ is the sigmoid function and, U_r and U_z are the weights from one hidden unit to its next hidden unit in the layer.

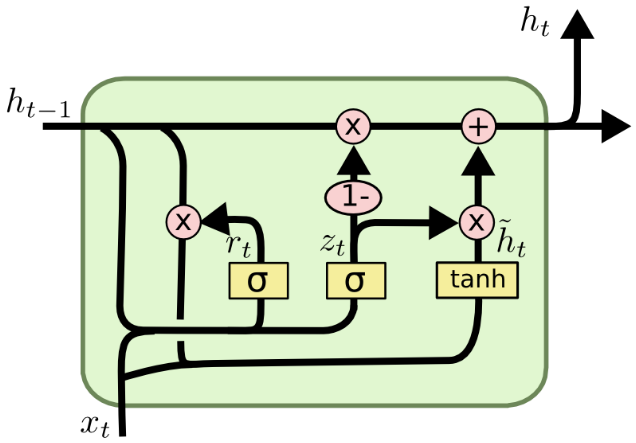

Fig. 1. A GRU cell with update (*z*) and reset (*r*) gates, and *h* as the activation and $\bar{h}$ as the candidate activation.

C. Bidirectional RNNs

One of the flaw with vanilla RNN is that it uses the past contexts only. Bidirectional RNNs (BRNNs) help us overcome this shortcoming by processing the data in both the forward and the backward direction in time before the layers output is fed to the output layer. Bidirectional RNNs calculate the forward and the backward hidden sequences, and the output sequence y by looping through the backward layer in time from n = T to 1 and the forward layer in time from n =1 to T. The output layer y_t is then updated as:

$$\overrightarrow{h_n} = \mathrm{F}(W_{x\vec{\mathrm{h}}} x_n + W_{\vec{\mathrm{h}}\vec{\mathrm{h}}} \overrightarrow{h_{n+1}} + b_{\vec{\mathrm{h}}}) \qquad (8)$$

$$\overleftarrow{h_n} = \mathrm{F}(W_{x\overleftarrow{h}} x_n + W_{\overleftarrow{h}\overleftarrow{h}} \overleftarrow{h_{n+1}} + b_{\overleftarrow{h}}) \qquad (9)$$

$$y_n = W_{\vec{\mathrm{h}}y} \overrightarrow{h_n} + W_{\overleftarrow{h}y} \overleftarrow{h_n} + b_y \qquad (10)$$

Bidirectional RNNs in combination with LSTM cells allows the architecture to access contexts in longer range in both the directions.

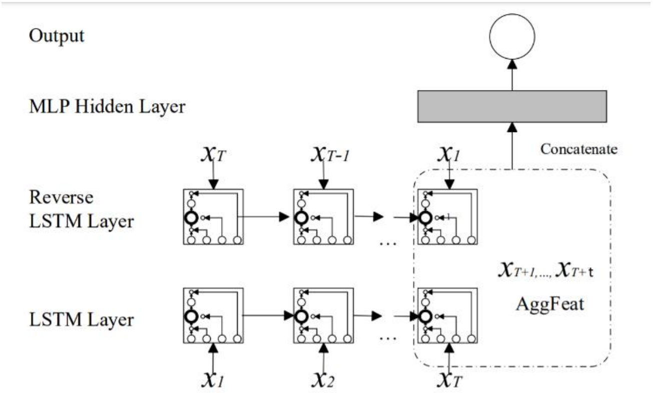

Fig. 2. A Bidirectional RNN with forward and backward layers.

D. Self-Attention

Starting from Bahdanau’s attention model [19] to the Transformer model [20] many attention models have been proposed in deep learning. The attention model allows the output to pay extra attention on the inputs while estimating the outpu. In contrast, the self-attention method allows interactions on inputs with each other i.e. this models allows calculation of the attention of all other inputs with respect to every single input. Text classification involves focusing on all the words. Therefore, given the requirement of our experiments we will be applying Lin et al., [21] self-attention mechanism proposed in 2017.

The h attention heads are utilized by the self-attention sub layers. To outline the sub layer abdicate, parameterized linear transformation is enforced to the concatenation of the outcome obtained from each head. Each attention head estimates a new sequence z = (z_1, z_2, ..., z_n) by operating on the input sequence x = (x_1, x_2, ..., x_n).

$$z_i = \sum_{k=1}^{n} \alpha_{ik}(x_k W^V) \qquad (11)$$

Each weight coefficient, α_{ik}, is estimated using a softmax function:

$$\alpha_{ik} = \frac{exp\ e_{ik}}{\sum_{j=1}^{n} exp\ e_{ik}} \qquad (12)$$

Further, by comparing the two inputs e_{ik} is calculated as:

$$e_{ik} = \frac{(x_i W^Q)\ (x_k W^j)^T}{\sqrt{d_z}} \tag{13}$$

W^V , W^Q, W^J ε $R^{d_x \times d_z}$ are the four matrix parameters. For every attention head and the layer, all these four matrices are always unique.

III. MODEL DETAILS

In this work, we have conducted experiments with two models: a GRU Classifier with Self-Attention (GRUSA) and a Bidirectional GRU Classifier with Self-Attention (BGRUSA).

GRUSA uses the vanilla Recurrent Neural Network with LSTM cells while BGRUSA uses the bidirectional LSTM network.

A. *GRU Classifier with Self-Attention*

GRUSA uses GRU cells in the Bidirectional Recurrent Neural Network architecture. At the top of this architecture the self-attention layer is implemented. This layer ensures that the model has better focus on all the words with respect to all the other words in the input.

B. *Bidirectional GRU Classifier with Self-Attention*

BGRUSA use the bidirectional approach wherein an RNN with GRU cells runs in forward direction and another RNN in backward direction. An RNN in both direction provides the extra context. Thus, producing an opportunity for better decision making. At the top of this architecture the self-attention layer is implemented. This layer ensures that the model has better focus on all the words with respect to all the other words in the input.

IV. EXPERIMENTS

A. *Dataset*

We have conducted training and testing of our models using the Employment Scam Aegean Dataset (EMSCAD) [6] which is a publicly available dataset containing 17,880 real-life job advertisements that aims at providing a clear picture of the Employment Scam problem to the research community. EMSCAD records were annotated by hand and were classified into two categories. The dataset contains 17,014 legitimate job advertisements and 866 fraudulent job advertisements. These advertisements were published between 2012 to 2014.

The dataset was divided into training, validation and testing sets randomly with 60% of real and fake records used for training, 20% for validation and 20% for testing.

B. *Implementation*

The training data is firstly preprocessed in order to prepare it for training the models. The string is encoded into utf-8 unicode. All the words are converted to lowercase. Porter stemmer is applied to the whole database to remove the common morphological and inflexional endings from the words. Several features from original dataset is removed. *job id* is removed since it has all the unique values. Further we have removed the columns where we have missing value in description column. Now we prepare two variations of this data. The first variation has *location, company profile, department, requirements, benefits, employment type, required experience, required education and industry* columns removed as these columns have more than 60% missing values. Further, we have added one column to the dataset by adding the length of description as an extra feature. The second variation of our dataset has several columns combined into one. The *description, location, department, company profile, requirements, benefits, employee type, required experience, required education, industry and function* columns combined into one feature. Hence, the available values that were removed in the first variation are added to the description column thereby adding more contexts for making the decision. Further, one-hot representation is used to represent both the variations of the dataset. The dataset thus obtained is used for training all the models. For RNN models based on first variation, the sentence length used in 1024 while for the second model the sentence length used is 2608. These values are decided based on the median length of the sentences.

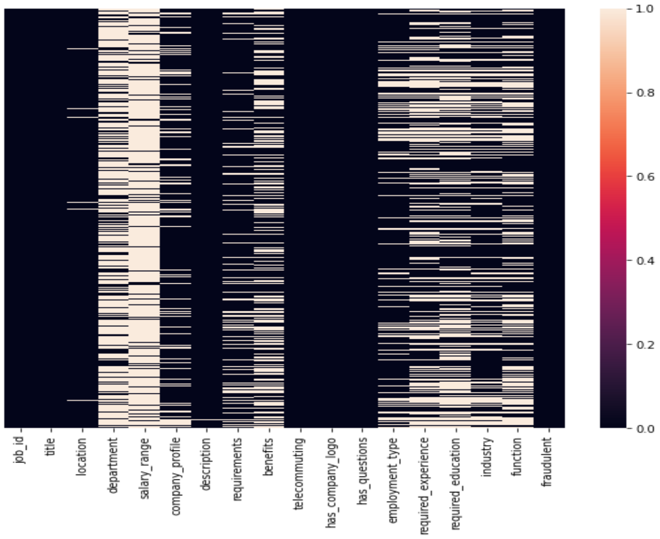

Fig. 3. Heatmap for null values in the dataset.

The GRUSA model, for the first variation of dataset, uses 1024 GRU cells followed by the self-attention layer to improve the learning over longer length. It is followed by a dense hidden layer of 2048 cells with sigmoid activation function which is further followed by the output layer which uses softmax as the activation function. Same configurations are used for the BGRUSA model as well. For the second variation of the data, we have used 2608 GRU cells followed by the self-attention model. The dense layer for this variation has 4096 cells with sigmoid activation function further followed by output layer with softmax function. Again, same configurations are used for BGRUSA model as well.

To compare our models learning, we have implemented other machine learning algorithms also. These algorithms include Logistic Regression, Decision Tree, Random Forest, XGBoost, K-Nearest Neighbor, Naïve Bayes and Support Vector Classifier. Besides these, we have also implemented the base GRU and Bidirectional GRU models without self-attention. We have used accuracy, precision, recall and F1 score to evaluate the learning of all the models.

All the models use binary cross entropy loss function, Adam to optimize the model and the learning rate of 0.004.

C. Results

The results of all the models are shared in the Table I on the next page. All the four GRU based models are able to learn the patterns better than all the other baselines models. Among baseline models KNN performs better than the other algorithms in terms of accuracy with 95.24% while XGBoost performs better in terms of F1 Score with 93.26%. The basic GRU model attains the accuracy of 93.01 and the F1 score of 94.25 while the bidirectional GRU (BGRU) model attains the accuracy of 94.62 and the F1 score of 94.76. The self-attention models perform better than non-attention models with GRUSA model attaining the accuracy of 95.49% and F1 Score of 94.98%. The bidirectional self-attention GRU model (BGRUSA) model outperforms all the models in terms of all the four metrics with the accuracy of 97.40 and the F1 score of 95.56.

TABLE I. RESULTS OF ALL THE MODELS

Models	**Accuracy**	**Precision**	**Recall**	**F1 Score**
GRU	93.01	85.36	95.82	94.25
BGRU	94.62	86.01	96.78	94.76
GRUSA	95.49	86.99	97.21	94.98
BGRUSA	97.40	88.90	98.38	95.56
Logistic Regression	83.23	78.98	86.33	84.76
Decision Tree	82.28	78.48	88.43	83.37
Random Forest	93.04	84.91	93.67	91.03
XGBoost	94.67	84.13	95.71	93.26
KNN (20n)	95.24	81.55	91.39	88.45
Naïve Bayes	94.03	78.01	85.41	79.60
SVC	79.10	75.58	86.27	80.57

D. Observations

Based on the results obtained from the set of experiments that we have conducted in this work, we come up with following observations:

- Bidirectional model performed better than the unidirectional architecture.
- Self-attention models perform better than the non-attention models.
- The bidirectional models are able to learn the sequences almost like the self-attention based unidirectional models.
- Balancing the dataset increases the overall performance of all the models.

V. CONCLUSION

In this paper, we have implemented a series of experiments with unidirectional and Bidirectional RNN architecture using GRU cell, firstly with self-attention and then without self-attention. Our results show that even the basic GRU model performs better than other baseline algorithms. The Bidirectional GRU is able to remember the text sequences better than the basic GRU model. When trained with self-attention, both the unidirectional and bidirectional models perform better than the non-attention models. The bidirectional self -attention model performs better than every other model for the task of fake job classification. In addition, resampling and balancing the dataset impacts the learning a lot and allows a more stable learning to take place.

REFERENCES

[1]. J. C. Chang and C.C. Lin, ”Recurrent-neural-network for language detection on Twitter code-switching corpus.” arXiv preprint arXiv:1412.4314 (2014).

[2]. L. Bing, and I Lane, ”Attention-based recurrent neural network models for joint intent detection and slot filling.” arXiv preprint arXiv:1609.01454 (2016).

[3]. P. Li, J. Li, F. Sun, and P. Wang, ”Short Text Emotion Analysis Based on Recurrent Neural Network.” In Proceedings of the 6th International Conference on Information Engineering, p. 6. ACM, 2017.

[4]. D. Tang, B. Qin, and T. Liu, ”Document modeling with gated recurrent neural network for sentiment classification.” In Proceedings of the 2015 conference on empirical methods in natural language processing, pp. 1422-1432. 2015.

[5]. A. Graves, A. R. Mohamed, and G. Hinton, ”Speech recognition with deep recurrent neural networks.” In Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, pp. 6645-6649. IEEE, 2013.

[6]. I. Sutskever, O. Vinyals, and Q. V. Le, ”Sequence to sequence learning with neural networks.” In Advances in neural information processing systems, pp. 3104-3112. 2014.

[7]. O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, ”Show and tell: A neural image caption generator.” In Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, pp. 3156-3164. IEEE, 2015.

[8]. S. Hochreiter, ”The vanishing gradient problem during learning recurrent neural nets and problem solutions.” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6, no. 02 (1998): 107-116.

[9]. R. Pascanu, T. Mikolov, and Y. Bengio, ”On the difficulty of training recurrent neural networks.” In International Conference on Machine Learning, pp. 1310-1318. 2013.

[10]. S. Hochreiter, and J. Schmidhuber, ”Long short-term memory.” Neural computation 9, no. 8 (1997): 1735-1780.

[11]. K. Cho, B. V. Merrinboer, 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).

[12]. O. Irsoy, and C. Cardie, ”Opinion mining with deep recurrent neural networks.” In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 720-728. 2014.

[13]. G. Pollastri, D. Przybylski, B. Rost, and P. Baldi, ”Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks

and profiles.” Proteins: Structure, Function, and Bioinformatics 47, no. 2 (2002): 228-235.

[14]. Y. Tang, and J. Liu. ”Gated Recurrent Units for Airline Sentiment Analysis of Twitter Data.”

[15]. G. Arevian, ”Recurrent neural networks for robust real-world text classification.” In Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence, pp. 326-329. IEEE Computer Society, 2007.

[16]. G. Mesnil, Y. Dauphin, K. Yao, Y. Bengio, L. Deng, D. H. Tur, X. He, ”Using recurrent neural networks for slot filling in spoken language understanding.” IEEE/ACM Transactions on Audio, Speech, and Language Processing 23, no. 3 (2015): 530-539.

[17]. J. Wang, L.C. Yu, K. R. Lai, and X. Zhang, ”Dimensional sentiment analysis using a regional CNN-LSTM model.” In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), vol. 2, pp. 225-230. 2016.

[18]. J. Y. Lee, and F. Dernoncourt, ”Sequential short-text classification with recurrent and convolutional neural networks.” arXiv preprint arXiv:1603.03827 (2016).

[19]. D. Bahdanau, K. Cho, and Y. Bengio, ”Neural machine translation by jointly learning to align and translate.” arXiv preprint arXiv:1409.0473 (2014).

[20]. Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. "Attention is all you need." In Advances in neural information processing systems, pp. 5998-6008. 2017.

[21]. Z. Lin, M. Feng, C. N. Santos, M. Yu, B. Xiang, B. Zhou, and Y. Bengio, ”A structured self-attentive sentence embedding.” arXiv preprint arXiv:1703.03130 (2017).