Check It Out : Politics and Neural Networks

by

Yash Kumar Lal, Dhruv Khattar, Vaibhav Kumar, Abhimanshu Mishra, Vasudeva Varma

in

Conference and Labs of the Evaluation Forum
(*CLEF-2018*)

Avignon, France

Report No: IIIT/TR/2018/-1

Centre for Search and Information Extraction Lab
International Institute of Information Technology
Hyderabad - 500 032, INDIA
September 2018

CEUR-WS.org/Vol-2125/paper_106.pdf

Check It Out : Politics and Neural Networks

Yash Kumar Lal, Dhruv Khattar, Vaibhav Kumar, Abhimanshu Mishra, and
Vasudeva Varma

[1] Manipal Institute of Technology, Manipal
yash.kumar4@learner.manipal.edu
[2] International Institute of Information Technology Hyderabad
dhruv.khattar@research.iiit.ac.in
[3] International Institute of Information Technology Hyderabad
vaibhav.kumar@research.iiit.ac.in
[4] Birla Institute of Technology, Mesra
abhimanshu@gmail.com
[5] International Institute of Information Technology Hyderabad
vv@iiit.ac.in [**]

Abstract. The task of fact-checking has been formalised as the assessment of the truthfulness of a claim. Be it a political proclamation or a technological development, verification of a new tidbit of information before its propagation to the general public is of utmost importance. Failing to do so leads to the spread of misinformation, which is a devious tool. Fact-checking is commonly performed by journalists, manually looking up information pertaining to the statement in question. This is a drawn out and tedious process with a chance of the concerned person not covering the domain exhaustively. Some of this effort is reduced by the use of knowledge bases created over a period of time. In this work under Task 2 (Factuality) of the CLEF 2018 CheckThat! Lab, we detail a neural network based methodology which models the textual data of a claim based on various representations of its words and characters. An affixed attention mechanism allows us to encapsulate linguistic features common in false claims. We achieve an accuracy of 39.57% on the task dataset.

Keywords: Fact Checking, Bidirectional LSTM, Attention Mechanism

1 Introduction

Fact-checking is the task of evaluating claims made in discourse by public figures like politicians, pundits etc. The process of creation and curation of an article often takes a journalist through the fact checking phase. This task is a part of the pipeline in the creation of knowledge bases for various domains. Apart from journalists, with the advent of the present political scenario, institutes and government organisations dedicated to fact-checking have risen to prominence.

[**] First four authors have equal contribution.

Lawyers, as part of the judicial process, are required to meticulously vet facts and statements before presenting it to the concerned authorities.

Even with the plethora of information available on the Internet, just a click away, it is a tedious and long drawn-out process. Generally, it involves first checking the validity of the claim from a curated list of trusted sources. Next, the consistency of the claim is verified by looking it up against various sources. For the preceding to happen, the fact-checker has to first fully dissect and understand the claim being made since there can be multiple interpretations of the same, or it can, in turn, be based on a combination of factors or pre-requisite claims. Websites providing this service generally provide in-depth analysis, encompassing the different possibilities, of a claim rather than a numeric score for its validity.

Fact-checking requires more research and a more advanced style of writing than ordinary journalism. The difficulty of fact-checking, exacerbated by a lack of resources for investigative journalism, leaves many harmful claims unchecked, particularly at the local level. Its effectiveness is negatively impacted by a gap in time and availability (defined as the necessity of a checker to have to look up a documented fact nevertheless).

The emergence of the field of computational journalism has promoted the conversation about the need and importance of automatic fact-checking systems. [1, 2] have pioneered the work being done to achieve the same. Bolstered by the advances being made in the domains of natural language processing, databases and information retrieval, the objective is to provide journalists with tools to effectively and accurately automate this element of their job. This automation is enabled by the increasing online availability of datasets, survey results, and reports in machine-readable formats by various institutions. Another layer of checks required is at the reader level.

With the advent and exponential rise of social media platforms, dissemination of news might wholly skip the traditional channel, thereby not being verified by journalists at all. Apart from this, an increase in citizen journalism [5] means that anyone is capable of creating and disseminating 'facts' without checks and consequences, leading to far-reaching misinformation.

Another problem with current fact-checking platforms is their outdated nature of publishing. Many of these rely on older content management systems built just for newspapers and blogs that are not designed for structured journalism. This limits how well they can be used in computational pipelines.

We propose a neural network based architecture for the task. We leverage the distributional semantics of the claim and model its temporal and sequential properties. The contribution of a word towards the validity of the claim is calculated in a differential manner since the output of the LSTM is passed into an attention layer [3], following which it goes through a dense layer and assigns an output class.

2 Related Work

Most of the work done towards accomplishment of this task requires manual effort. [7] formalised the definition of the task at hand and proposed an approach and format of constructing the required dataset. [8] theorise the construction of an ideal fact-checking system, having built ClaimBuster, a system to detect if a sentence has a claim in it and if the claim is worth validating. Such a tool could streamline the task for journalists, causing them to do research sparingly instead of looking at all sentences in discourse. Due to the prominence of fake news in the media nowadays, organisations such as FullFact have taken it upon themselves to weed out the problem, setting out a roadmap to do the same and inviting collaboration in the field. They have released a survey detailing the techniques being adopted now, possible ways of tackling this problem, and defining global standards to be followed for the creation of an effective, combined solution. [6] developed sentence representations from natural language inference data which can be used as part of a transfer learning approach to solve this problem. FullFact has hypothesised that these vectors can act as effective inputs to a bi-LSTM max-pooling network.

3 Model Architecture

We now describe our approach to developing a fact-checking system and the reasons behind devising such a model. We start with an explanation of the type of embeddings we have used and proceed to describe our proposed model, a bidirectional long short term memory network with an attention mechanism. An overview of the architecture can be seen in Figure 1. Finally, we cover how the parameters are learned.

Distributed Word Embeddings

Considering the effectiveness of distributional semantics in modeling language data, we use a pre-trained 300 dimensional Word2Vec [4] model trained over 100 billion words in the Google News corpus using the Continuous Bag of Words architecture. These map the words in a language to a high dimensional real-valued vectors to capture hidden semantic and syntactic properties of words, and are typically learned from large, unannotated text corpora. For each word in the title, we obtain its equivalent Word2Vec embeddings using the model described above.

We now move on to describing the crux of our proposed approach.

Bidirectional LSTM with Attention

Recurrent Neural Network (RNN) is a class of artificial neural networks which utilizes sequential information and maintains history through its intermediate layers. A standard RNN has an internal state whose output at every time-step which can be expressed in terms of that of previous time-steps. However, it has been seen that standard RNNs suffer from a problem of vanishing gradients [9]. This means it will not be able to efficiently model dependencies and interactions between words that are a few steps apart. LSTMs are able to tackle this issue

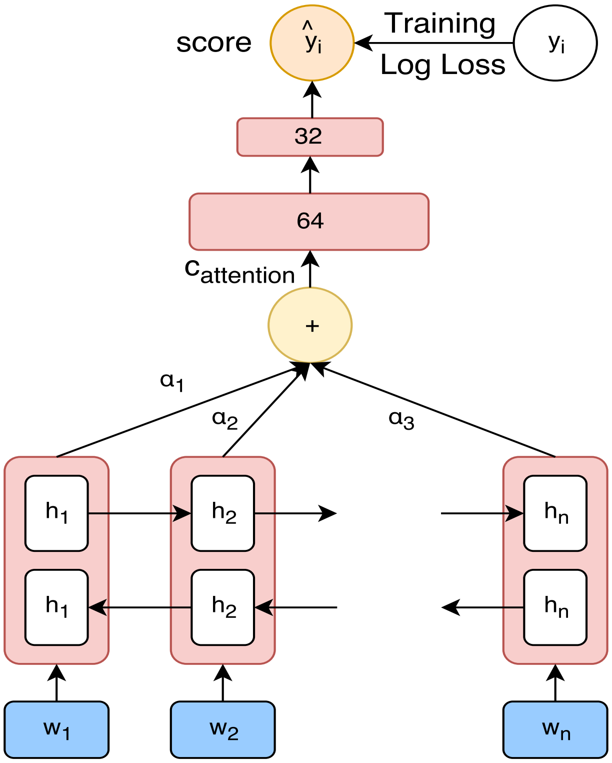

Fig. 1. Overview of proposed model

by their use of gating mechanisms. We convert the words of the claim into the previously mentioned types of embeddings to act as input to our bidirectional LSTMs.

$(\overrightarrow{h}_1, \overrightarrow{h}_2, \ldots, \overrightarrow{h}_R)$ represent forward states of the LSTM and its state updates satisfy the following equations:

$$[\overrightarrow{f_t}, \overrightarrow{i_t}, \overrightarrow{o_t}] = \sigma[\overrightarrow{W}[\overrightarrow{h}_{t-1}, \overrightarrow{r_t}] + \overrightarrow{b}] \tag{1}$$

$$\overrightarrow{l_t} = \tanh[\overrightarrow{V}[\overrightarrow{h}_{t-1}, \overrightarrow{r_t}] + \overrightarrow{d}] \tag{2}$$

$$\overrightarrow{c_t} = \overrightarrow{f_t} \cdot \overrightarrow{c}_{t-1} + \overrightarrow{i_t} \cdot \overrightarrow{l_t} \tag{3}$$

$$\overrightarrow{h_t} = \overrightarrow{o_t} \cdot \tanh(\overrightarrow{c_t}) \tag{4}$$

here σ is the logistic sigmoid function, $\overrightarrow{f_t}$, $\overrightarrow{i_t}$, $\overrightarrow{o_t}$ represent the forget, input and output gates respectively. $\overrightarrow{r_t}$ denotes the input at time t and $\overrightarrow{h_t}$ denotes the latent state, $\overrightarrow{b_t}$ and $\overrightarrow{d_t}$ represent the bias terms. The forget, input and output gates control the flow of information throughout the sequence. $\overrightarrow{W}$ and $\overrightarrow{V}$ are matrices which represent the weights associated with the connections.

$(\overleftarrow{h}_1, \overleftarrow{h}_2, \ldots, \overleftarrow{h}_R)$ denote the backward states and its updates can be computed similarly.

The number of bidirectional LSTM units is set to a constant K, which is the maximum length of all title lengths of records used in training. The forward and backward states are then concatenated to obtain $(h_1, h_2, \ldots, h_K)$, where

$$h_i = \overrightarrow{h}_i \overleftarrow{h}_i \tag{5}$$

Finally, we are left with the task of figuring out the significance of each word in the sequence i.e. how much a word shapes towards a writing style characteristic of factual claims. The effectiveness of attention mechanisms have been proven for the task of neural machine translation [3] and it has the same effect in this case. The goal of attention mechanisms in such tasks is to derive context vectors which capture relevant source side information and help predict the current target word. The sequence of annotations generated by the encoder to come up with a context vector capturing how each word contributes to the record's factuality is of paramount importance to this model. In a typical RNN encoder-decoder framework [3], a context vector is generated at each time-step to predict the target word. However, we only need it for calculation of context vector for a single time-step.

$$c_{attention} = \sum_{j=1}^{K} \alpha_j h_j \tag{6}$$

where, h_1,. . . ,h_K represents the sequence of annotations to which the encoder maps the post title vector and each α_j represents the respective weight corresponding to each annotation h_j.

Learning the Parameters

We use binary cross-entropy as the loss optimization function for our model. The cross-entropy method [14] is an iterative procedure where each iteration can be divided into two stages:

(1) Generate a random data sample (vectors, trajectories etc.) according to a specified mechanism.

(2) Update the parameters of the random mechanism based on the data to produce a "better" sample in the next iteration.

4 Evaluations

4.1 Dataset

[10] has provided a compilation of all the sentences from the first and second of the 2016 US Presidential election debates and the Vice-Presidential debate. The discourse in these moderated debates contains a plethora of claims made by each candidate as they attempted to make their case to hold the highest office in the United States.

The first distinction between the types of sentences is that some of them contain claims, and need to be processed by the model, while the others do not, and can be ignored. Further, each of the sentences that has been tagged as containing a claim is assigned a label pertaining to the validity of the claim. Each claim can be one of three types : TRUE, HALF-TRUE and FALSE.

Each record is associated with a line number (position in discourse of the debate), the speaker (which can be either candidate, or the moderator, or SYSTEM in case of external noise), a tag establishing whether it is a claim or not, a chronological claim number, and a label for the validity of the claim. In case of the sentence not containing a claim, it is considered part of a separate 'N/A' class. Table 1 shows some basic statistics found on analysis of the training data files.

Metric	Prez Debate 1	Prez Debate 2	Vice Prez Debate
Number of sentences	1403	1303	1358
Words in longest claim	64	91	91
Chars in longest claim	372	506	511

Table 1. Basic statistics of training data files

4.2 Experiment Settings

We combine the three files and then randomly split the 4064 sentences into training and validation set in a 4:1 ratio. This ensures that the two sets do not overlap. The model hyperparameters are tuned over the validation set. We initialise the fully connected network weights with the uniform distribution in the range $-\sqrt{6/(fanin + fanout)}$ and $\sqrt{6/(fanin + fanout)}$ [12]. We used a batch size of 256 and adadelta [13] as a gradient based optimizer for learning the parameters of the model.

4.3 Results

The model was evaluated over test set of 7 files which contained a combined total of 192 possible claims to be verified. The subsequent result files were submitted to the CLEF Fact Checking Lab [10] to be evaluated against a variety of metrics. Table 2 details the outcome of the evaluation.

Model	Accuracy	Macro Recall
BiLSTM with Attention	39.57%	0.33
Team BigIR	39.57%	0.33
Team FACTR	41.01%	0.36
Team UPV-INAOE	38.85%	0.34

Table 2. Model Evaluation

We reach a comparable accuracy to various proposed approaches to the task.

The use of separate knowledge bases and other information sources would have contributed to higher accuracy or better recall for other participants' models [11] in this task.

5 Conclusion and Future Work

We see that LSTMs are able to model claims and figure out possible dependencies between its constituents. Furthermore, the use of an attention layer allows our approach to capture features unique to fake claims and leverage them in the verification of others.

As part of improving this approach, we would like to try the addition of a knowledge base for political claims, and augmenting the dataset with tweets from Politico's Twitter handle tagged with #PresidentialDebate. It is also possible to generate a similarity score between a claim to be verified and each fact stored in the knowledge base, and evaluate if they refer to the same core idea.

References

1. Cohen, S. and Li, C. and Yang, J. and Yu, C.: Computational Journalism: A Call to Arms to Database Researchers. In: Proceedings of the Conference on Innovative Data Systems Research (2011)
2. Flew, T. and Daniel, A. and Spurgeon, C.L.: The promise of computational journalism. In: Proceedings of the Australian and New Zealand Communication Association Conference (2012)
3. Bahdanau, D and Cho, K. and Bengio, Y.: Neural machine translation by jointly learning to align and translate. (2014)
4. Mikolov, T. and Chen, K. and Corrado, G. and Dean, J.: Efficient Estimation of Word Representations in Vector Space. (2013)
5. Goode, L.: Social news, citizen journalism and democracy. In: New Media & Society (2009)
6. Conneau, A. and Kiela, D. and Schwenk, H. and Barrault, L. and Bordes, A.: Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing (2017)
7. Vlachos, A. and Reidel, S.: Fact checking: Task definition and dataset construction. In: Proceedings of the ACL Workshop on Language Technologies and Computational Social Science (2014)
8. Hassan, N. and Adair, B. and Hamilton, J. and Li, C. and Tremayne, M. and Yang, J. and Yu, C.: The Quest to Automate Fact-Checking. In: Proceedings of the Computation + Journalism Symposium (2015)
9. Hochreiter, S. and Schmidhuber, J.: Long short-term memory. In: Neural Computation (1997)
10. Nakov, Preslav and Barrón-Cedeño, Alberto and Elsayed, Tamer and Suwaileh, Reem and Màrquez, Lluís and Zaghouani, Wajdi and Atanasova, Pepa and Kyuchukov, Spas and Da San Martino, Giovanni: Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims.

In: Proceedings of the Ninth International Conference of the CLEF Association: Experimental IR Meets Multilinguality, Multimodality, and Interaction

11. Barrón-Cedeño, Alberto and Elsayed, Tamer and Suwaileh, Reem and Màrquez, Lluís and Atanasova, Pepa and Zaghouani, Wajdi and Kyuchukov, Spas and Da San Martino, Giovanni and Nakov, Preslav: Overview of the CLEF-2018 Lab on Automatic Identification and Verification of Claims in Political Debates, Task 2: Factuality. In: CLEF 2018 Working Notes. Working Notes of CLEF 2018 - Conference and Labs of the Evaluation Forum
12. Glorot, X. and Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (2010)
13. Zeiler, M. D.: ADADELTA: an adaptive learning rate method. In: arXiv preprint arXiv:1212.5701 (2012)
14. de Boer, P. and Kroese, D.P. and Mannor, S. and Rubinstein, R.Y.:A Tutorial on the Cross-Entropy Method. In: Annals of Operations Research (2005)