3HAN: A Deep Neural Network for Fake News Detection

Sneha Singhania(✉), Nigel Fernandez, and Shrisha Rao

International Institute of Information Technology - Bangalore, Bangalore, India
{sneha.a,nigelsteven.fernandez}@iiitb.org, shrao@ieee.org

Abstract. The rapid spread of fake news is a serious problem calling for AI solutions. We employ a deep learning based automated detector through a three level hierarchical attention network (3HAN) for fast, accurate detection of fake news. 3HAN has three levels, one each for words, sentences, and the headline, and constructs a news vector: an effective representation of an input news article, by processing an article in an hierarchical bottom-up manner. The headline is known to be a distinguishing feature of fake news, and furthermore, relatively few words and sentences in an article are more important than the rest. 3HAN gives a differential importance to parts of an article, on account of its three layers of attention. By experiments on a large real-world data set, we observe the effectiveness of 3HAN with an accuracy of 96.77%. Unlike some other deep learning models, 3HAN provides an understandable output through the attention weights given to different parts of an article, which can be visualized through a heatmap to enable further manual fact checking.

Keywords: Fake news · Deep learning · Text representation · Attention mechanism · Text classification

1 Introduction

The spread of fake news is a matter of concern due to its possible role in manipulating public opinion. We define fake news in line with The New York Times as a "made up story with the intention to deceive, often with monetary gain as a motive" [1]. The fake news problem is complex given its varied interpretations across demographics.

We present a three level hierarchical attention network (3HAN) which creates an effective representation of a news article called *news vector*. A news vector can be used to classify an article by assigning a probability of being fake. Unlike other neural models which are opaque in their internal reasoning and give results that are difficult to analyze, 3HAN provides an importance score for each word and sentence of an input article based on its relevance in arriving at the output probability of that article being fake. These importance scores can be visualized

S. Singhania and N. Fernandez—These authors contributed equally to this work.

D. Liu et al. (Eds.): ICONIP 2017, Part II, LNCS 10635, pp. 572–581, 2017.
https://doi.org/10.1007/978-3-319-70096-0_59

through a heatmap, providing key words and sentences to be investigated by human fact-checkers.

Current work in detecting misinformation is divided between automated fact checking [2], reaction based analysis [3] and style based analysis [4]. We explore the nascent domain of using neural models to detect fake news. Current state-of-the-art general purpose text classifiers like Bag-of-words [5], Bag-of-ngrams with SVM [6], CNNs, LSTMs and GRUs [7] can be used to classify articles by simply concatenating the headline with the body. This concatenation though, fails to exploit the article structure.

In 3HAN, we interpret the structure of an article as a three level hierarchy modelling article semantics on the principle of compositionality [8]. Words form sentences, sentences form the body and the headline with the body forms the article. We hypothesize forming an effective representation of an article using the hierarchy and the interactions between its parts. These interactions take the form of context of a word in its neighbouring words, coherence of a sentence with its neighbouring sentences and stance of a headline with respect to the body. Words, sentences and headline are differentially informative dependent on their interactions in the formation of a news vector. We incorporate three layers of attention mechanisms [9] to exploit this differential relevance.

The design of 3HAN is inspired by the hierarchical attention network (HAN) [10]. HAN is used to form a general document representation. We design 3HAN unique to the detection of fake news. When manually fact-checking an article the first thing that catches the eye is the headline. We observe a headline to be (i) a distinctive feature of an article [11], (ii) a concise summary of the article body and (iii) inherently containing useful information in the form of its stance with respect to the body. We refer to these observations as our *headline premise*. The third level in 3HAN is especially designed to use our headline premise.

From our headline premise, we hypothesize that a neural model should accurately classify articles based on headlines alone. Using this hypothesis, we use headlines to perform a supervised pre-training of the initial layers of 3HAN for a better initialization of 3HAN. The visualization of attention layers in 3HAN indicates important parts of an article instrumental in detecting an article as fake news. These important parts can be further investigated by human fact-checkers.

We compare the performance of 3HAN with multiple state-of-the-art traditional and neural baselines. Experiments on a large real world news data set demonstrate the superior performance of 3HAN over all baselines with 3HAN performing with an accuracy of 96.24%. Our pre-trained 3HAN model is our best performing model with an accuracy of 96.77%.[1]

2 Model Design

The architecture of 3HAN is shown in Fig. 1. We define a news vector as a projection of a news article into a vector representation suitable for effective

[1] Our code is available at: https://github.com/ni9elf/3HAN.

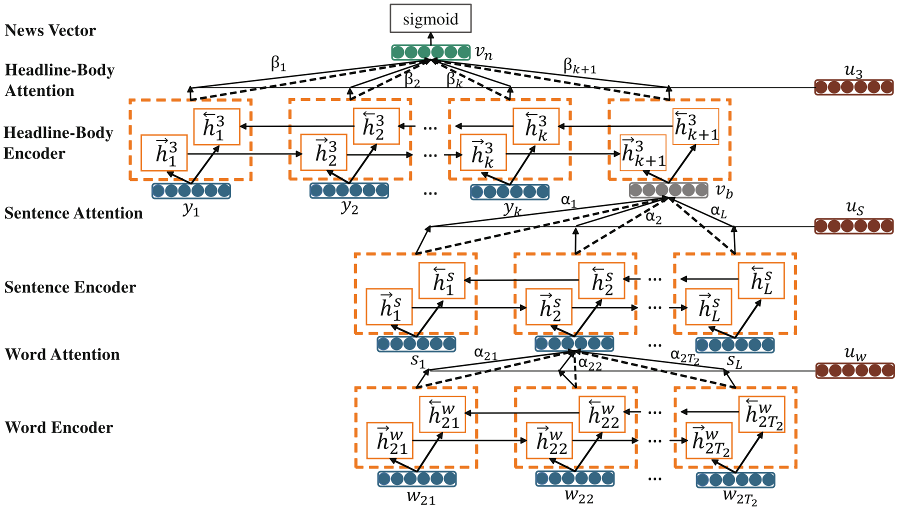

Fig. 1. Model Architecture of 3HAN

classification of articles. A news vector is constructed using 3HAN. To capture the body hierarchy and interactions between parts when forming the news vector, 3HAN uses the following parts from HAN [10]: word sequence encoder, word level attention (Layer 1), sentence encoder, sentence level attention (Layer 2). In addition to the preceding parts, we exploit our headline premise by adding: headline-body encoder and headline-body level attention (Layer 3).

Sequence Encoder using GRU. A Gated Recurrent Unit (GRU) [12] adaptively captures dependencies between sequential input sequences over time. Gating signals control how the previous hidden state h_{t-1} and current input x_t generate an intermediate hidden state $\widetilde{h_t}$ to update the current hidden state h_t. GRU consists of a reset gate r_t and an update gate z_t. r_t determines how to combine x_t with h_{t-1} while z_t determines how much of h_{t-1} and $\widetilde{h_t}$ to use. $\odot$ denotes the Hadamard product. The GRU model is presented at time t as:

$$\widetilde{h_t} = \tanh\left(W_h x_t + U_h\left(r_t \odot h_{t-1}\right) + b_h\right) \tag{1}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \widetilde{h_t} \tag{2}$$

with the gates presented as:

$$z_t = \sigma\left(W_z x_t + U_z h_{t-1} + b_z\right),\ r_t = \sigma\left(W_r x_t + U_r h_{t-1} + b_r\right) \tag{3}$$

Word Encoder. We denote word j of sentence i by w_{ij} with sentence i containing T_i words. Each word w_{ij} is converted to a word embedding x_{ij} using GloVe [13] embedding $W_e\left(x_{ij} = W_e\left(w_{ij}\right)\right)$. We use a bidirectional GRU [9] to form an annotation of each word which summarizes the *context* of the word with preceding and following words in the sentence. A bidirectional GRU consists of a forward $\overrightarrow{\text{GRU}}$ and backward $\overleftarrow{\text{GRU}}$. The overhead arrow in our notation does not denote a vector, it instead denotes the direction of the GRU run. $\overrightarrow{\text{GRU}}$ reads the word embedding sequence ordered $(x_{i1}, x_{i2}, \ldots, x_{iT_i})$ to form forward annotations using hidden states $\left(\overrightarrow{h}^w_{i1}, \overrightarrow{h}^w_{i2}, \ldots, \overrightarrow{h}^w_{iT_i}\right)$. Similarly $\overleftarrow{\text{GRU}}$ reads the word embedding sequence ordered $(x_{iT_i}, x_{iT_i-1}, \ldots, x_{i1})$ to form backward annotations $\left(\overleftarrow{h}^w_{iT_i}, \overleftarrow{h}^w_{iT_i-1}, \ldots, \overleftarrow{h}^w_{i1}\right)$. h^w_{ij} is formed as $\left[\overrightarrow{h}^w_{ij}, \overleftarrow{h}^w_{ij}\right]$ (concatenation).

$$\overrightarrow{h}^w_{ij} = \overrightarrow{\text{GRU}}\left(x_{ik}\right), k \in [1, j] \tag{4}$$

$$\overleftarrow{h}^w_{ij} = \overleftarrow{\text{GRU}}\left(x_{ik}\right), k \in [T_i, j] \tag{5}$$

$$h^w_{ij} = \left[\overrightarrow{h}^w_{ij}, \ \overleftarrow{h}^w_{ij}\right] \tag{6}$$

Word Attention. A sentence representation is formed using an attention layer to extract relevant words of a sentence. The word annotation h^w_{ij} is fed through a one-layer MLP to get a hidden representation u_{ij} [10]. The similarity of each word u_{ij} with a *word level relevance vector* u_w decides the attention weights α_{ij} normalized using a softmax function [10]. The sentence encoding s_i is a weighted attentive sum of the word annotations. The relevance vector can be interpreted as representing the contextually most relevant word over all words in the sentence. u_w is fixed over all inputs as a global parameter of our model and jointly learned in the training process.

$$u_{ij} = \tanh\left(W_w h^w_{ij} + b_w\right) \tag{7}$$

$$\alpha_{ij} = \frac{\exp\left(u^T_{ij} u_w\right)}{\sum_j \exp\left(u^T_{ij} u_w\right)}, \ s_i = \sum_j \alpha_{ij} h^w_{ij} \tag{8}$$

Sentence Encoder. Similar to the word encoder, a bidirectional GRU is applied to $(s_1, s_2, \ldots, s_L)$ to compute the forward annotations $\overrightarrow{h}^s_i$ and backward annotations $\overleftarrow{h}^s_i$ for each sentence. These annotations capture the *coherence* of a sentence with respect to its neighbouring sentences in both directions of the body. h^s_i is formed as $\left[\overrightarrow{h}^s_i, \overleftarrow{h}^s_i\right]$.

Sentence Attention. Similar to word attention, we identify relevant sentences in the formation of the body vector v_b by using an attention layer. A *sentence level relevance vector* u_s decides attention weights α_i for sentence annotation h^s_i. u_s can be interpreted as representing the coherently most relevant sentence over all sentences in the body. v_b is composed using $\sum_i \alpha_i h^s_i$.

Headline Encoder. To exploit our headline premise we design a third layer of encoding and attention with the headline being inputted word by word. We denote the k words of the headline by w_{01} to w_{0k}. The word embedding y_i for word w_{0i} is obtained using GloVe embeddings (W_e) by $y_i = W_e\left(w_{0i}\right)$. We denote v_b as y_{k+1}. A bidirectional GRU is run on $(y_1, y_2, \ldots, y_{k+1})$ to compute the forward and backward annotations of each word. These annotations capture the *stance of the headline* words with respect to the body word. The digit 3 in our notation denotes the third level. h_i^3 is formed as $\left[\overrightarrow{h}_i^3, \overleftarrow{h}_i^3\right]$.

$$\overrightarrow{h}_i^3 = \overrightarrow{\text{GRU}}\left(y_j\right), j \in [1, i], \ \overleftarrow{h}_i^3 = \overleftarrow{\text{GRU}}\left(y_j\right), j \in [k+1, i] \tag{9}$$

Headline Attention. A *relevance vector* u_3 is used to compute the attention weights β_i for annotation h_i^3. The news vector v_n is formed as the weighted sum of the annotations h_i^3 with β_i as the weights.

$$u_i = \tanh\left(W_3 h_i^3 + b_3\right) \tag{10}$$

$$\beta_i = \frac{\exp\left(u_i^T u_3\right)}{\sum_i \exp\left(u_i^T u_3\right)}, \ v_n = \sum_i \beta_i h_i^3 \tag{11}$$

News Vector for Classification. We use the news vector v_n as a feature vector for classification. We use the sigmoid layer $z = \text{sigmoid}\left(W_c v_n + b_c\right)$ as our classifier with binary cross-entropy loss $L = -\sum_d p_d \log q_d$ to train 3HAN. In the loss function q_d is the predicted probability and p_d is the ground truth label (either fake or genuine) of article d.

Supervised Pre-training using Headlines We propose a supervised pre-training of Layer 1 consisting of the word encoder and an attention layer of 3HAN for a better initialization of the model. The pre-training is performed using the headlines only. The output label for a headline input is the corresponding article label.

3 Experiments

3.1 News Data Set

Due to the high turnaround time of manual fact-checking, the number of available manually fact-checked articles is too few to train deep neural models. We shift our fact-checked requirement from an article level to a website level. Keeping with our definition of fake news, we assume that every article from a website shares the same label (fake or genuine) as its containing website. PolitiFact [14] a respected fact-checking website released a list of sites manually investigated and labelled. We use those sites from this list labelled fake. Forbes [15] compiled a list of popular genuine sites across US demographics. Statistics of our data set is provided in Table 1. To maintain a similar distribution as fake articles, we use genuine articles from January 1, 2016 to June 1, 2017, with 65% coming from the 2016 US elections and politics, 15% from world news, 15% from regional news and 5% from entertainment.

3.2 Baselines

To validate the effectiveness of our model, we compare 3HAN with current state-of-the-art traditional and deep learning models. The input is the article text formed by concatenating the headline with the body.

Table 1. Dataset Statistics: (average words per sentence, average sentences per article)

Type	Sites	Articles	Average Words	Average Sentences
Fake	19	20,372	34.20	16.44
Genuine	9	20,932	32.78	27.55

Word Count Based Models. These methods use a hand crafted feature vector derived from variations of frequency of words of an article. A binomial logistic regression is used as the classifier.

1. *Majority* uses the heuristic of taking the majority label in the training set as the assigning label to every point in the test set.
2. *Bag-of-words and its TF-IDF* constructs a vocabulary of the most frequent 50,000 words [5]. The count of these words is used as features. The TF-IDF count is used as features in the other model variant.
3. *Bag-of-ngrams and its TF-IDF* uses the count of the 50,000 most frequent ngrams ($n <= 5$). The features are formed as in the previous model.
4. *SVM+Bigrams* uses the count of the 50,000 most frequent bigrams as features with an SVM classifier [6].

Neural Models. The classifier used is a dense sigmoid layer.

1. *GloVe-Ave* flattens the article text to a word level granularity as a sequence of words. The GloVe embeddings of all words are averaged to form the feature vector.
2. *GRU* treats the article text as a sequence of words. A GRU with an annotation dimension of 300 is run on the sequence of GloVe word embeddings. The hidden annotation after the last time step is used as the feature vector.
3. *GRU-Ave* runs a GRU on the sequence of word embeddings and returns all hidden annotations at each time step. The average of these hidden annotations is used as the feature vector.
4. *HAN and Variants* include HAN-Ave, Han-Max and HAN [10]. HAN uses a two level hierarchical attention network. HAN-Ave and Han-Max replaces the attention mechanism with average and max pooling for composition respectively. Since the code is not officially released we use our own implementation.

3.3 Experimental Settings

We split sentences of bodies and tokenized sentences and headlines into words using Stanford CoreNLP [16]. We lower cased and cleaned tokens by retaining alphabets, numerals and significant punctuation marks. When building the vocabulary we retained words with frequency more than 5. We treat words appearing exactly 5 times as a special single unknown token (UNK). We used 100 dimensional GloVe embeddings to initialize our word embedding matrix and allowed it to be fine tuned. For missing words in GloVe, we initialized their word embedding from a uniform distribution on $(-0.25, 0.25)$ [17].

We padded (or truncated) each sentence and headline to an average word count of 32 and each article to an average sentence count of 21. Hyper parameters are tuned on the validation set. We used 100 dimensional GloVe embeddings and 50 dimensional GRU annotations giving a combined annotation of 100 dimensions. The relevance vector at word, sentence and headline-body level are of 100 dimensions trained as a parameter of our model. We used SGD with a learning rate of 0.01, momentum of 0.9 and mini batch size of 32 to train all neural models. Accuracy was our evaluation metric since our data set is balanced.

3.4 Results and Analysis

We used a train, validation and test split of 20% | 10% | 70% for neural models and a train and test split of 30% | 70% for word count based models. In 3HAN-Ave vectors are composed using average, in 3HAN-Max vectors are composed using max pooling, 3HAN is our proposed model with an attention mechanism for composition and 3HAN+PT denotes our pre-trained 3HAN model. Results are reported in Table 2 and demonstrate the effectiveness of 3HAN and 3HAN+PT due to their best performance over all models.

Neural models using the hierarchical structure (HAN and variants, 3HAN and variants) give a higher accuracy than other baselines. The attention mechanism is a more effective composition operator than average or max pooling.

Table 2. Accuracy in Article Classification as Fake or Genuine

Word Count Based Models

Model	**Accuracy**
Majority	49.42%
Bag-of-words	90.21%
Bag-of-words +TFIDF	91.92%
Bag-of-ngrams	91.41%
Bag-of-ngrams +TFIDF	92.47%
SVM+Bigrams	83.12%

Neural Network Models

Model	**Accuracy**
GloVe-Ave	93.63%
GRU	91.11%
GRU-Ave	95.65%
HAN-Ave	94.91%
HAN-Max	94.66%
HAN	95.4%
3HAN-Ave	94.81%
3HAN-Max	95.25%
3HAN	**96.24%**
3HAN+PT	**96.77%**

This is demonstrated by the higher accuracy of 3HAN against 3HAN-Ave and 3HAN-Max. Our headline premise is valid since 3HAN which devotes a separate third level in the hierarchy for the headline performs better than HAN. HAN is indifferent to the headline and focuses its two hierarchical levels only on words and sentences. Pre-training helps in better initialization of 3HAN with 3HAN+PT outperforming 3HAN.

4 Discussion and Insights

The visualization of attention layers provides evidence. An advantage of attention based neural models is the visualization of attention layers which provides insight into the internal classification process. On the other hand, non-attention based models work like a black box. 3HAN provides attention weights to words, sentences and headline of an article. These attention weights are useful for further human fact-checking. A human fact-checker can focus on verifying sentences with high attention weights. Similarly, words with high attention weights can be investigated for inaccuracies.

We visualize the attention weights given to words, sentences and the headline for a sample article through a heatmap in Fig. 2. The sentences with the top five attention weights and the first eight words in each sentence are shown for clarity. Word attention weights α_w are normalized using sentence attention weights α_s by $\alpha_w = \sqrt{\alpha_s}\alpha_w$. Sentence attention weights are shown on the extreme left edge. We observe that sentence 5 and has been assigned the highest weight (0.287). Interestingly, sentence 5 which states "Even refugee welcoming Canada levies a 12 percent penalty on immigrant money" is a factually incorrect sentence.

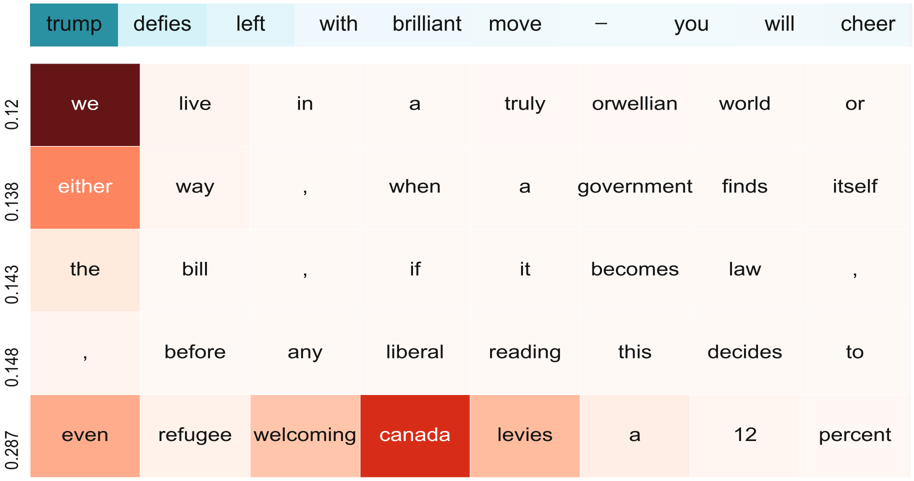

Fig. 2. Visualization of Attention Layers in a Fake News Article with Headline "Trump Defies Left with Brilliant Move - You Will Cheer"

Word count based models perform well. The high accuracy of simple word count based models which do not take into account word ordering or semantics is an indication of vocabulary and patterns of word usage from the vocabulary being a distinguishing feature between fake news and true news.

The attention mechanism is effective. This is observed through the superior performance of HAN compared to non-attention based 3HAN-Max and 3HAN-Ave.

Our headline premise is valid. This is observed from the superior performance of 3HAN to HAN with the third hierarchical level of 3HAN especially designed for our headline premise playing a role.

The inverted pyramid style of writing is used. Inverted pyramid refers to distributing information in decreasing importance in an article. We inferred the usage of the inverted pyramid through our experiments from the small improvement in accuracy even with higher padding sentence counts. Fake news articles tend to be repetitive in information content [11].

5 Conclusion and Future Work

In this paper, we presented 3HAN which creates news vector, an effective representation of an article for detection as fake news. We demonstrated the superior accuracy of 3HAN over other state-of-the-art models. We highlighted the use of visualization of the attention layers. We plan to deploy a web application based on 3HAN which provides detection of fake news as a service and learns in a real time online manner from new manually fact-checked articles.

Acknowledgements. We thank the anonymous ICONIP reviewers as well as G. Srinivasaraghavan, Shreyak Upadhyay and Rishabh Manoj for their helpful comments.

References

1. Tavernisen, S.: As fake news spreads lies, more readers shrug at the truth. New York Times, 6 December 2016. http://nyti.ms/2lw56HN
2. Vlachos, A., Riedel, S.: Identification and verification of simple claims about statistical properties. In: 20th Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pp. 2596–2601, September 2015. doi:10.18653/v1/d151312
3. Acemoglu, D., Ozdaglar, A., ParandehGheibi, A.: Spread of (mis) information in social networks. Games Econ. Behav. **70**(2), 194–227 (2010)
4. Afroz, S., Brennan, M., Greenstadt, R.: Detecting hoaxes, frauds, and deception in writing style online. In: 33rd IEEE Symposium on Security and Privacy (SP 2012), pp. 461–475. IEEE, May 2012
5. Joachims, T.: Text categorization with Support Vector Machines: Learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998). doi:10.1007/BFb0026683

6. Wang, S., Manning, C.D.: Baselines and bigrams: Simple, good sentiment and topic classification. In: 50th Annual Meeting of the Association for Computational Linguistics (ACL 2012), pp. 90–94, July 2012
7. Tang, D., Qin, B., Liu, T.: Document modeling with gated recurrent neural network for sentiment classification. In: 20th Conference on Empirical Methods in Natural Language Processing (EMNLP 2015), pp. 1422–1432, September 2015
8. Frege, G.: Sense and reference. Philos. Rev. **57**(3), 209–230 (1948)
9. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations (ICLR 2015), May 2015
10. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2016), pp. 1480–1489, June 2016
11. Horne, B., Adali, S.: This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In: Workshop of the 11th International AAAI Conference on Web and Social Media (ICWSM 2017), May 2017
12. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. In: 19th Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1724–1734, October 2014
13. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: 19th Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1532–1543, October 2014
14. Gillin, J.: Politifact's guide to fake news websites and what they peddle. PunditFact, 20 April 2017. http://bit.ly/2pHYKDV
15. Glader, P.: 10 journalism brands where you find real facts rather than alternative facts. Forbes, 1 February 2017. http://bit.ly/2sXPpvf
16. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014), pp. 55–60, June 2014
17. Kim, Y.: Convolutional neural networks for sentence classification. In: 19th Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 1746–1751, October 2014