Sentiment and Semantic Deep Hierarchical Attention Neural Network for fine grained News Classification

by

Sriteja Allaparthi, Ganesh Yaparla, Vikram Pudi

in

IEEE International Conference on Big Knowledge
(*ICBK-2018*)

SIngapore

Report No: IIIT/TR/2018/-1

Centre for Data Engineering
International Institute of Information Technology
Hyderabad - 500 032, INDIA
November 2018

Sentiment and Semantic Deep Hierarchical Attention Neural Network for fine grained News Classification

Sri Teja Allaparthi, Ganesh Yaparla, Vikram Pudi
Data Sciences and Analytics Center(DSAC),
Kohli Center on Intelligent Systems,
International Institute of Information Technology-Hyderabad,
Hyderabad, India.
{sriteja.allaparthi, ganesh.yaparla}@research.iiit.ac.in, vikram@iiit.ac.in

***Abstract*—The purpose of this study is to examine the differences between different types of news stories. Given the huge impact of social networks, online content plays an important role in forming or changing the opinions of people. Unlike traditional journalism where only certain news organizations can publish content, online journalism has given chance even for individuals to publish. This has its own advantages like individual empowerment but has given a chance to a lot of malicious entities to spread misinformation for their own benefit. As reported by many organizations in recent history, this even has influence on major events like the outcome of elections. Therefore, it is of great importance now, to have some sort of automated classification of news stories. In this work, we propose a deep hierarchical attention neural architecture combining sentiment and semantic embeddings for more accurate fine grained classification of news stories. Experimental results show that the sentiment embedding along with semantic information outperform several state-of-the-art methods in this task.**

***Index Terms*—News Classification, Sentiment Analysis, Hierarchical Attention Network, Fake News Detection.**

I. Introduction

Due to the rise of online social media, it has become the dominant channel for the spread of news. It provides the mechanism to publish news events as they happen. As soon as the information is uploaded to the web, it is available around the globe. In these times where blogging is considered as journalism and thousands of websites are being built daily, it is unfortunate that the authenticity of the news on a topic can be undermined.

Fake news is a term often used to refer to fabricated news, which has no basis, but presented as being factually accurate. In some some cases, what appears to be fake news may in fact be a news satire, which uses exaggeration and introduces non-factual elements that are intended to amuse or make a point rather than to deceive. Hence, if we deeply observe, we can find different kinds of news.

An analysis by the Internet media company, BuzzFeed revealed that during the final three months of 2016 US Presidential campaign, the 20 most popular false election stories generated around 1.3 million more Facebook engagements in the form of shares, likes, comments than did the 20 most popular legitimate stories [1]. One of the most popular fake story was "Pope Francis shocks world, endorses Donald Trump for President". Fake news can distort people's beliefs even after being debunked [2]. People are more inclined to believe that the information is true if they are encountered it before. Hence, all these have huge impact on the society and there is a serious need to classify different types of news stories to make people know about what they are reading. Such system can be a bulwark to the dissemination of fake, false, conspiratorial and misleading news. And recently in 2017, UK House of Commons conducted a parliamentary inquiry into the "growing phenomenon of fake news" [3].

If we come to the basic point of how these news are being disseminated in the social networks, then that would help us to reduce it. The basic concept of social network is to follow the people you like or believe or look up to. If something comes from such people in your network then there is a fair chance of you believing that story if you are a little aware of that before. Here, we certainly should worry about people (including journalists) unwittingly sharing misinformation, but far more concerning are the systematic disinformation campaigns. Previous attempts to influence public opinion relied on 'one-to-many' broadcast technologies but, social networks allow 'atoms' of propaganda to be directly targeted at users who are more likely to accept and share a particular message. Once they inadvertently share a misleading or fabricated article, image, video or meme, the next person who sees it in their social feed probably trusts the original poster, and goes on to share it themselves. These 'atoms' then rocket through the information ecosystem at high speed powered by trusted peer-to-peer networks.

On the whole, each individual is playing a crucial role in the ecosystem. Every time we passively accept information without double-checking, or share a post, image or video before we have verified it, we are adding indirectly contributing for the dissemination of misinformation. The ecosystem is now so polluted, we have to take responsibility for independently checking what we see online. Hence, there is a great need for an automated intelligent system which can classify what stories people are reading to so that they can verify before believing or sharing such news stories.

In this paper, we propose a deep hierarchical attention neural architecture combining sentiment and semantic embeddings for fine grained classification of news stories. We also show experimentally that our algorithm performs successfully on the available news articles from OpenSources dataset [4], which is a curated resource for assessing online information sources, available for public use.

The rest of the paper is organized as follows: Section 2 gives a brief overview on the related work. Section 3 describes the proposed approach. Section 4 presents experiments and results. Section 5 presents conclusion and future work.

II. Related Work

A review of previous literature reveals a series of works aiming at the task of fake news detection, clickbait identification, classification of satire and true content has been studied separately. Recently many methods and studies have been put forward for fake news detection. All these studies mainly tries to analyze the content of news articles. BD Horne et al. (2017) studies the stylistic, complexity and psychological features [5]. Studies include the behavioural patterns by exploiting supervised learning techniques [6], [7], linguistic based features to identify Clickbaits [8], [9], deceptive writing styles in [10], [11]. SVM and other handcrafted features are used by [12]–[17]. Many studies have been done related to social networks analyzing the automatic response detection [21]–[24] and considering other hand crafted features like Facebook likes, shares and comments [25]–[31]. All these approaches are restricted by using these features, fail to capture the complex structure of the content in news articles. Deep learning have been used by [18]–[20], tries to capture the information with in the content of news articles.

Attention mechanism was fist used in machine translation [32]. In image caption generation, attention is used by [33]. In many other applications like parsing [39], question answering [34]–[37], relation classification [38] and document classification [41] attention is used. The document structure is exploited mainly by 3HAN [40] which is inspired from HAN [41]. While HAN uses a hierarchical attention neural network for document classification, [40] specifically focuses on fake news detection. Our proposed model has some overlap with these works. However, these studies do not explicitly targeted fine grained news classification at large scale. In our work, we mainly concentrate on fine grained classification of news articles which includes 9 different classes other than "fake". Furthermore, we hope that this type of in depth classification systems can be helpful for news organizations to automatically tag articles and for users to tag or know about a news story before they mistook and share with in their social networks believing it as a true story.

III. Model

In this section, we give details of the proposed model, which we call SSD-HAN, Sentiment and Semantic Deep Hierarchical Neural Network. The architecture of SSD-HAN is shown in "Figs. 1 to 3". The model consists of two main parts, a module for extracting the sentiment embedding of news articles, and a module for representing the contextual information. We construct the news vector by using both the sentiment embedding vector and contextual information vector of a news article which helps in classification. A news vector is generated using SSD-HAN. To capture the information from large document, SSD-HAN uses a Hierarchical Attention Network which is inspired from HAN [41]. HAN contains following layers: word sequence encoder, word level attention (Layer 1), sentence encoder, sentence level attention (Layer 2). This network captures the contextual information of an article. In addition to these, we use another Hierarchical Attention Network which captures the sentiment embedding of the article. We combine these two, sentiment embedding and semantic embedding of a document along with the title of news article to construct a news vector. We describe the details of different components in the following sections.

A. GRU based Sequence Encoder

Gated Recurrent Units (GRUs) [43] are improved version of standard Recurrent Neural Networks(RNNs) [44]. GRU uses update gate and reset gate which helps in solving the vanishing gradient problem in standard RNNs. Basically, these two gates are two vectors which decides what information should be passed to the output.

1) Update Gate: Update Gate helps the model to determine how much of the information from the previous words needs to be passed along to the future time steps. That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem [42].

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{1}$$

z_t decides how much the unit updates its activation.

2) Reset Gate: Reset Gate is used to decide how much of the past information to forget. We use the reset gate to store the relevant information from the past.

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{2}$$

r_t allows how much the unit considers the previously computed state. If r_t is close to zero then the whole past information is discarded.

3) Final Memory at current time step: This is a vector which holds information for the current unit and passes it down to the network. Update gate is needed in calculating this. It determines what to collect from the current memory content and what from the previous steps.

$$\widetilde{h}_t = \tanh\left(W x_t + U(r \odot h_{t-1})\right) \tag{3}$$

$$h_t = (1 - z_t) h_{t-1} + z_t \widetilde{h}_t \tag{4}$$

h_t is the final activation of a GRU unit at time t.

B. News article representation

We focus on fine-grained news classification in this work. Assume that a news article has a Title T_i, a Body which consists of L sentences s_i and each sentence having N_i words. w_{in} with $n \in [1, N]$ represents the words in the i^{th} sentence L_i. Our model, SSD-HAN, represents the news article into a news vector on which we perform classification. In the following sections, we discuss how we construct the final news vector capturing both the sentiment and the semantic information from word vectors using SSD-HAN.

C. Word Encoder

Given a set of words W_i, forming a sentence i in a news article, we convert them into word embeddings using GloVe [46]. Using these vectors as inputs to the network, we try to capture the information around each word in both the directions. We use a bi-directional GRU to capture the contextual information around each word in both directions. The forward GRU reads each word in a sentence starting from first word, while the backward GRU reads words starting from last. For each word w_{in}, it's word annotation contains both the annotations given by forward and backward GRUs $[\overrightarrow{h_{in}}, \overleftarrow{h_{in}}]$. We use attention mechanism over these word annotations which gives different scores for each annotation proportional to relevance of the word in meaning of the sentence. We discuss this in detail in the below Attention section. We use all these word annotations along with attention to get a sentence vector s_i.

$$\overrightarrow{h_{in}} = \overrightarrow{GRU}(x_{in}), n \in [1, N] \tag{5}$$

$$\overleftarrow{h_{in}} = \overleftarrow{GRU}(x_{in}), n \in [N, 1] \tag{6}$$

$$h_{in} = [\overrightarrow{h_{in}}, \overleftarrow{h_{in}}] \tag{7}$$

$\overrightarrow{h_{in}}$ is the annotation given by forward GRU and $\overleftarrow{h_{in}}$ is by backward GRU. h_{in} is the final word annotation generated by concatenating $\overrightarrow{h_{in}}$ and $\overleftarrow{h_{in}}$.

D. Sentence Encoder

Similar to the word encoder, we use another bi-directional GRU which takes sentence vectors as input. For each sentence, its annotation contains both the annotations given by forward and backward GRUs $[\overrightarrow{h_{in}}, \overleftarrow{h_{in}}]$. We use attention mechanism on these sentence annotations to capture the relevant sentences which mainly contribute to the meaning of entire body of a news article. Using these sentence annotations along with attention, we get a body vector b.

$$\overrightarrow{h_i} = \overrightarrow{GRU}(s_i) \tag{8}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(s_i) \tag{9}$$

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{10}$$

$\overrightarrow{h_i}$ is the annotation given by forward GRU and $\overleftarrow{h_i}$ is by backward GRU. h_i is the final sentence annotation generated by concatenating $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$.

E. Attention

An attention model [45] takes n arguments h_1, ..., h_n, and a context c. It returns a vector which is supposed to be the summary of h_i, focusing on information linked to the context c. More formally, the representation of next layer in the network, will be the weighted sum of the input arguments according to their relevance for the classification task.

1) Word Attention: In our case, first the bi-directional GRUs give the word annotations on which we apply attention mechanism. Not all the words contribute in equal to the meaning of a sentence. Let H be the matrix having vectors $[\overrightarrow{h_{in}}, \overleftarrow{h_{in}}]$, of all the words in a sentence. H is given as input to the Attention model which gives a vector output, which is the weighted sum of vectors in H. The weights are proportional to the relevance of that word to the information in given classification task. c_w is a trained parameter vector which helps in deciding the weights according to the relevance.

$$a_{in} = \tanh(W_w h_{in}) \tag{11}$$

$$\alpha_{in} = \frac{\exp(a_{in} c_w)}{\sum_n \exp(a_{in}^T c_w)} \tag{12}$$

$$s_i = \sum_n \alpha_{in} h_{in} \tag{13}$$

c_w is the context vector initialized at random and learned in training, s_i is the sentence vector formed by attention mechanism which aggregates the representation of the informative words in sentence L_i.

2) Sentence Attention: Similar to word attention, we identify relevant sentences in the formation of the body vector b by using an attention layer.

$$a_i = \tanh(W_s h_i) \tag{14}$$

$$\alpha_i = \frac{\exp(a_i c_s)}{\sum_i \exp(a_i^T c_s)} \tag{15}$$

$$b = \sum_i \alpha_i h_i \tag{16}$$

c_s is context vector initialized at random and learned in training, b is the body vector that summarizes all the information of sentences in a news article.

F. Body Sentiment embedding - Ave

The network shown in "Fig. 1" trained on sentiment classification task, gives the sentiment embedding of news article. In this network, we use word encoder, word attention, sentence encoder and sentence attention layers to capture polarity of the body of a news article. This network is trained separately using two datasets namely IMDB taken from [50], and Yelp15 from [49]. We use these datasets to maximum utilize the vocabulary available. We construct the final sentiment embedding by taking the average of these two embeddings.

$$b_{si} = HAN_s(x_{in}) \tag{17}$$

$$b_s = Avg(b_{si}) \tag{18}$$

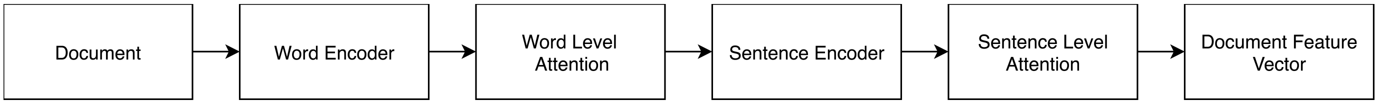


Fig. 1. HAN - Hierarchical Attention Neural Network Architecture.

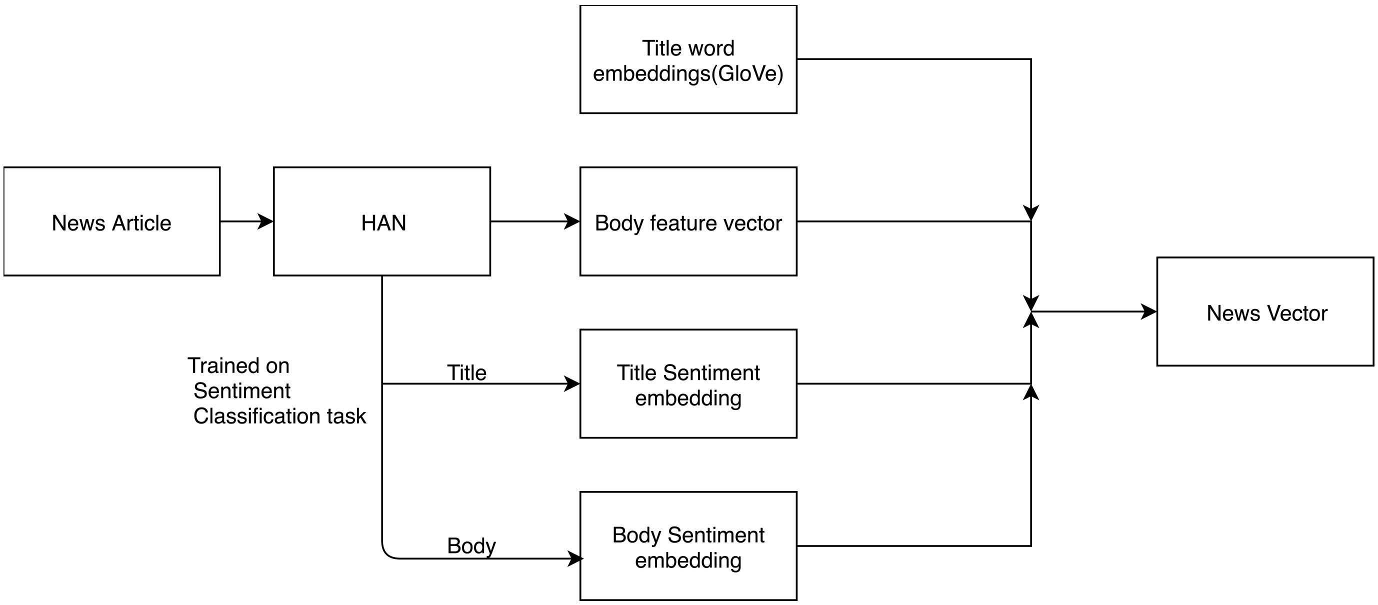


Fig. 2. SSD-HAN - Sentiment and Semantic Deep Hierarchical Attention Neural Network Architecture.

b_{si} is the sentiment embedding of body for each dataset. b_s is generated by taking average of these b_{si}. HAN_s represents the sentiment classification task by HAN.

G. Title Sentiment embedding - Ave

Similar to body sentiment embedding, words from the title are taken as input to network by converting them into word embeddings using GloVe and the sentiment embedding is generated for the title of a news article. Average of the two embeddings from IMDB, Yelp15 is considered. As titles are a strong differentiating factor between fake and real news [**?**], we hypothesize that titles play an important role in fine-grained classification of news articles as well.

$$t_{si} = HAN_s(t_n) \tag{19}$$

$$t_s = Avg(t_{si}) \tag{20}$$

t_{si} is the sentiment embedding of title for each dataset. t_s is generated by taking average of these t_{si}. HAN_s represents the sentiment classification task by HAN.

H. News Article Encoder

The body vector b, title sentiment embedding t_s, body sentiment embedding b_s and words in title are given as input to GRU. This annotation captures the context and polarity of headline with respect to body of the news article.

$$x = [x_{tn}, b, ts, bs] \tag{21}$$

$$\overrightarrow{h_i} = \overrightarrow{GRU}(x_i) \tag{22}$$

$$\overleftarrow{h_i} = \overleftarrow{GRU}(x_i) \tag{23}$$

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \tag{24}$$

where x is the concatenation of GloVe vectors of words in title, body vector, title sentiment embedding and body sentiment embedding.

I. News Article Attention

In this section, final news vector is constructed as the weighted sum of the output vectors from the News Article Encoder relevant to given classification task.

$$a_i = \tanh(Wh_i) \tag{25}$$

$$\alpha_i = \frac{\exp(a_i c_n)}{\sum_i \exp(a_i^T c_n)} \tag{26}$$

$$n_v = \sum_i \alpha_i h_i \tag{27}$$

c_n is the context vector initialized at random and learned in training, n_v is the final news vector formed by attention mechanism.

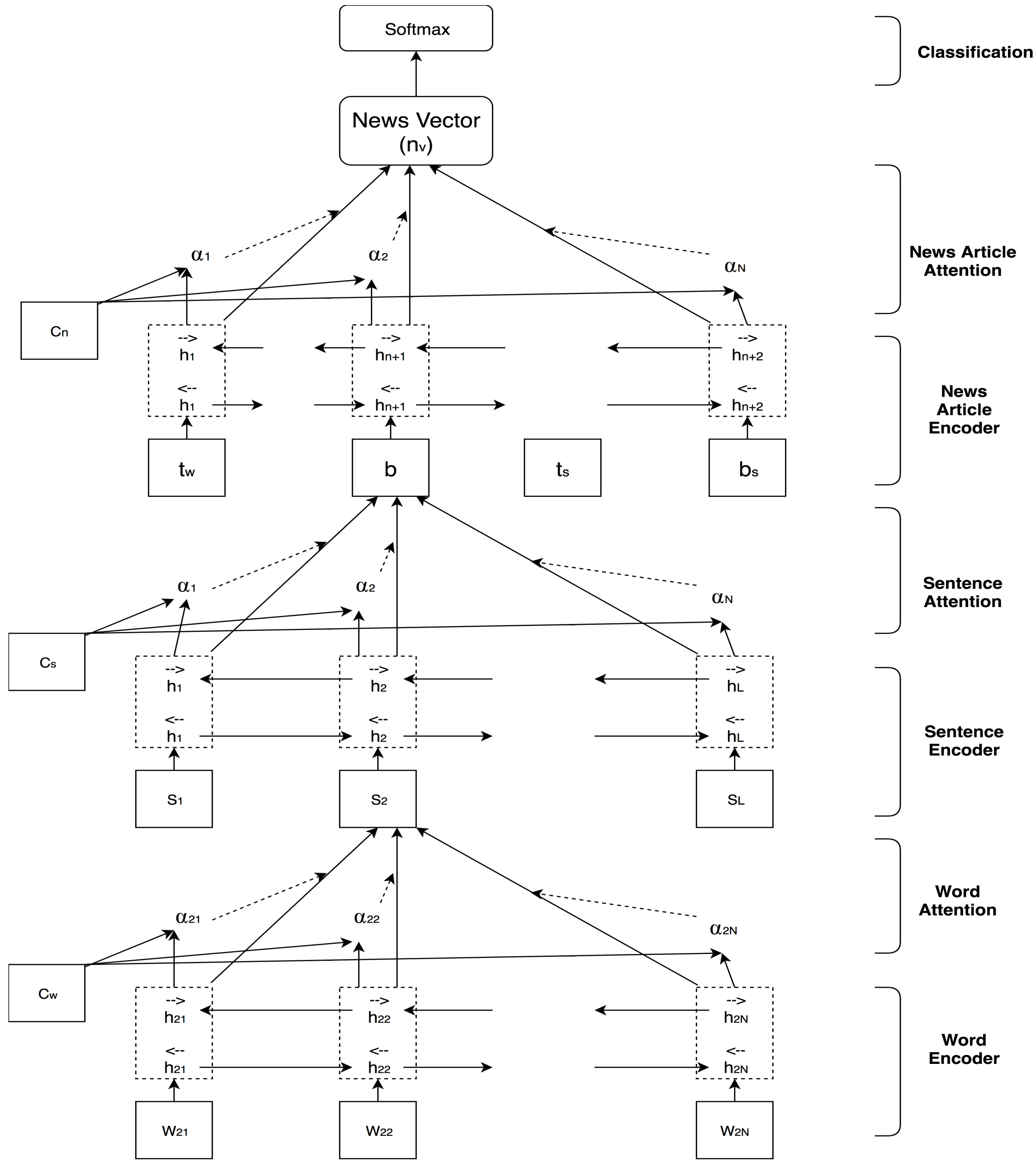


Fig. 3. SSD-HAN - Sentiment and Semantic Deep Hierarchical Attention Neural Network Architecture.

J. News Article classification

The final news vector n_v is used as a feature vector for document classification. We use negative log likelihood as the loss function for training.

$$p = softmax(W_f n_v) \tag{28}$$

$$L = -\sum_{i \in A} \sum_{c \in C} \log p_{ic} \tag{29}$$

A is the set of articles and C is the set of classes. p_{ic} is the model probability of assigning class c to article i.

IV. Experiments

In this section, we demonstrate the quality of SSD-HAN on OpenSources Dataset [4]. In the main set of experiments, we evaluate the accuracy of the classification produced by SSD-HAN.

A. Dataset

We evaluate the effectiveness of our model on a large scale OpenSources dataset. OpenSources is a curated resource for assessing online information sources. Websites in this resource range from credible news sources to misleading and outright fake websites. We consider a subset of it which contains a total of 29,65,603 articles across all the classes to maintain balance among the classes in the dataset. A summary of the statistics is listed in Table I. The datasets used to capture sentiment embeddings are Yelp15 and IMDB. We have taken the the following tags from OpenSources dataset as classes.

1) Fake
2) Satire
3) Bias
4) Conspiracy
5) Rumor
6) Hate
7) Clickbait
8) Political
9) Reliable
10) Unreliable

TABLE I
DATASET

Tag	***Number of news articles***
Fake	4,28,083
Satire	1,46,080
Bias	4,00,444
Conspiracy	4,05,981
Hate	1,17,374
Clickbait	2,92,201
Political	4,35,471
Reliable	4,20,139
Unreliable	3,19,830

B. Baselines

We compare SSD-HAN with several baseline methods including traditional approaches such as word count based models and neural network models.

1) Bag of Words: In this we selected, 50,000 most frequent words from vocabulary of the articles in the training set. The frequency of each word is used to calculated the tf-idf scores which are used as features and logistic regression is used for the classification of news articles.

2) N-grams: Similar to the bag of words, in this, we selected 2,00,000 most frequent N-grams ($N <= 5$). The frequency of each N-gram is used to calculate the tf-idf scores which are used as features.

3) SVM with N-grams: Similar to the above two methods, in this, we selected the 2,00,000 most frequent n-gram tf-idf scores as features and SVM is used for classification.

4) GloVe-Ave: In this, entire news content along with the title is considered as a sequence of words. GloVe vectors are considered for all these words. All these word embeddings are averaged to construct the final news vector. A fully connected layer with sigmoid activation is used for classification.

5) GRU-Ave: Similar to the above method, in this, we considered news body along with the title as sequence of words. GloVe vectors are considered for all these words, given as inputs for the GRU. The annotations given by GRU are averaged to construct the final news vector. A fully connected layer with sigmoid activation is used for classification.

6) HAN: This is a two layer hierarchical attention neural network considering word level and sentence level attention for document classification.

7) 3HAN: This is a three layer hierarchical attention neural network considering word, sentence and headline-body level attention specifically designed for fake news detection.

C. Model Settings

We use Stanford CoreNLP [47] for splitting text into sentences and for tokenization of words. All the words are preprocessed by considering alphabets, numbers and significant punctuation marks. All the words are converted into lower case for uniformity. All the words that occur less than 6 times are discarded from the vocabulary.

We consider maximum sentence count to be considered from an article as 20 and maximum word count per sentence as 15. We use 50 dimensional GRU annotations, which give 100 dimension embedding, by combining forward and backward GRU annotations for words and sentences. We initialize the word and sentence level context vectors at random, which also have a dimension of 100 are learned by the model in training. We use 200 dimensional GloVe word embeddings to represent words as input to the network. For missing words in GloVe, we initialize their word embedding from a uniform distribution on $(-0.25, 0.25)$, as reported in [48] that it works better than initializing it to zeros. Stochastic gradient descent is used for training all the models with a mini-batch size of 64, learning rate as 0.01 and momentum as 0.9.

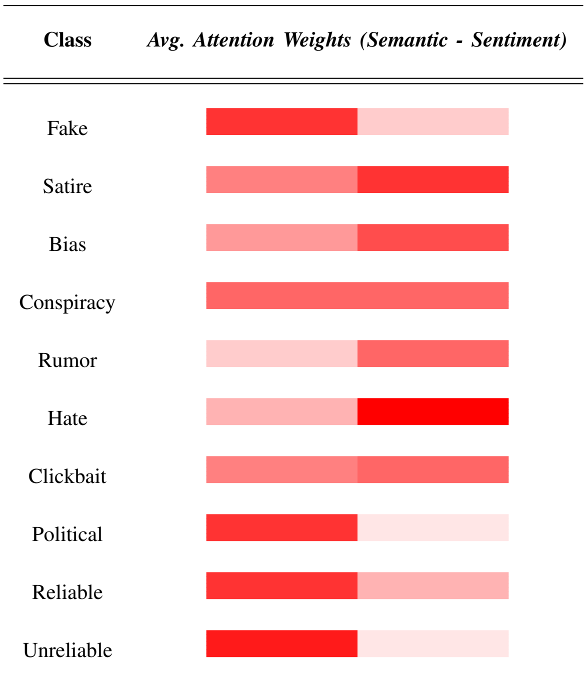
D. Results and Discussion

The experimental results of all the compared models are shown in Table III. Results show that our model, SSD-HAN gives the best performance on the OpenSources dataset in classification task. We use a train and test split of 70%, 30% for word count based models and train, validation and test split of 70%, 10%, 20% for neural network models. As the dataset is balanced, we use accuracy as a metric to evaluate all the models.

From Table IV we can see that the neural network models not using attention mechanism performs better that traditional baseline models while the models that use attention mechanism outperforms them. And finally, SSD-HAN considering both sentiment embedding and contextual information performs better than all these methods.

The inverted style of writing news articles as mentioned by [5], [40] is also exploited by our model. This refers to information represented in decreasing importance. Considering the sentiment embedding along with contextual information further helps in fine-grained classification of news articles. This is demonstrated by the higher accuracy of SSD-HAN.

TABLE II
ATTENTION WEIGHTS CONSIDERING BOTH SENTIMENT AND SEMANTIC EMBEDDINGS



In order to validate that our model exploits the sentiment and contextual information for fine grained classification of news articles, we visualize the final layer attention weights for all the classes. We show how the weights vary between the sentiment and contextual information by considering the average of weights on a sample of 10,000 articles from each class. All the weights from the semantic embeddings are averaged across one article and further averaged across the sample of articles and similarly with sentiment embedding. The intensity of colour denotes the weightage given to sentiment and contextual information. From the results in Table II, it is evident that the news articles from the classes Fake, Political, Reliable and Unreliable are highly dependent on contextual information, while Satire, Bias, Conspiracy, Rumor and Hate are dependent on both information for classification.

V. CONCLUSION AND FUTURE WORK

In this work, we study the fine grained classification of news articles. While the existing work has typically addressed the problem by focusing on individual classification tasks. We propose a model, Sentiment and Semantic Deep Hierarchical Neural Network (SSD-HAN) for fine grained news classification. Our model progressively builds the news vector by giving importance to informative words, sentences along with their sentiment embedding using attention mechanism. We demonstrated the performance of SSD-HAN over traditional word count based and standard neural network models. In future, we want to explore more about the hidden patterns in articles that contributed more towards classification and visualizing those patterns and features in a more comprehensive manner.

TABLE III
WORD COUNT BASED MODELS

Methods	***Accuracy***
Bag of Words	68.67
Bag of Words + TF-IDF	69.34
Bag of N-grams	72.04
Bag of N-grams + TF-IDF	73.42
SVM + N-grams	65.20

TABLE IV
NEURAL NETWORK MODELS

Methods	***Accuracy***
GloVe-Ave	78.67
GRU-Ave	81.34
HAN	81.78
3HAN	85.48
SSD-HAN	**89.5**

REFERENCES

[1] https://www.buzzfeed.com/craigsilverman/viral-fake-election-news-outperformed-real-news-on-facebook

[2] J. de Keersmaecker, Roets, A. (2017). Fake news: Incorrect, but hard to correct. The role of cognitive ability on the impact of false information on social impressions. Intelligence, 65, 107110. doi: 10.1016/j.intell.2017.10.005

[3] http://www.bbc.com/news/uk-politics-38790254

[4] http://www.opensources.co/

[5] Horne, Benjamin D., and Sibel Adali. "This just in: fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news." arXiv preprint arXiv:1703.09398 (2017).

[6] Antoniadis, Sotirios, Iouliana Litou, and Vana Kalogeraki. "A model for identifying misinformation in online social networks." OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, Cham, 2015.

[7] M. Rajdev and K. Lee, Fake and spam messages: Detecting misinformation during natural disasters on social media, in Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on, vol. 1.IEEE, 2015, pp. 1720.

[8] Johannes FuAlrnkranz. 1998. A study using n-gram features for text categorization. Austrian Research Institute for ArtiinAcal Intelligence 3 (1998), 1aAS10.

[9] Niall J Conroy Yimin Chen and Victoria L Rubin. 2015. Extracting and Composing Robust Features with Denoising Autoencoders. In In Proceedings of the 2015 ACM on Workshop on Multimodal Deception Detection (ACM 15). ACM, 1519.

[10] William Yang Wang. 2017. "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. CoRR abs/1705.00648 (2017). arXiv:1705.00648 http://arxiv.org/abs/1705.00648

[11] Michael Brennan Sadia Afroz and Rachel Greenstadt. 2012. Detecting hoaxes, frauds, and deception in writing style online. In ISSP (2012).

[12] Victoria L. Rubin Conroy, Niall J. and Yimin Chen. 2015. "Automatic deception detection: Methods for nding fake news.". Proceedings of the Association for Information Science and Technology (2015), 14.

[13] Veronica Perez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2017. Automatic Detection of Fake News. CoRR abs/1708.07104 (2017). arXiv:1708.07104

[14] Benjamin Riedel, Isabelle Augenstein, Georgios P. Spithourakis, and Sebastian Riedel. 2017. A simple but tough-to-beat baseline for the Fake News Challenge stance detection task. CoRR abs/1707.03264 (2017). arXiv:1707.03264 http://arxiv.org/abs/ 1707.03264

[15] Victoria L Rubin. 2016. "Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News.". Proceedings of NAACL-HLT (2016).

[16] Kai Shu. 2017. "Fake News Detection on Social Media: A Data Mining Perspective.". ACM SIGKDD Explorations Newsletter 19.1 (2017), 223.

[17] Tacchini, Eugenio, et al. "Some like it hoax: Automated fake news detection in social networks." arXiv preprint arXiv:1704.07506 (2017).

[18] Ruchansky, Natali, Sungyong Seo, and Yan Liu. "CSI: A Hybrid Deep Model for Fake News Detection." Proceedings of the 2017 ACM on Conference on Information and Knowledge Management. ACM, 2017.

[19] Yu, Feng, et al. "A convolutional approach for misinformation identification." Proceedings of the 26th International Joint Conference on Artificial Intelligence. AAAI Press, 2017.

[20] Shaurya Rohatgi, Rajeev Bhatt Ambati, and Neisarg Dave. 2018. Fake News Detection. In Proceedings of ACM Conference (IST 597: Principles of Arti cial Intelligence). ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/ nnnnnnn.nnnnnnn

[21] Adrien Friggeri, Lada A Adamic, Dean Eckles, and Justin Cheng. 2014. Rumor Cascades.. In ICWSM.

[22] Fang Jin, Edward Dougherty, Parang Saraf, Yang Cao, and Naren Ramakrishnan. 2013. Epidemiological modeling of news and rumors on twitter. In Proceedings of the 7th Workshop on Social Network Mining and Analysis. ACM, 8.

[23] Srijan Kumar, Robert West, and Jure Leskovec. 2016. Disinformation on the web: Impact, characteristics, and detection of wikipedia hoaxes. In Proceedings of the 25th International Conference on World Wide Web. International World Wide Web Conferences Steering Committee, 591602.

[24] Kate Starbird, Jim Maddock, Mania Orand, Peg Achterman, and Robert M Mason. 2014. Rumors, false flags, and digital vigilantes: Misinformation on twitter after the 2013 boston marathon bombing. iConference 2014 Proceedings (2014).

[25] Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In Proceedings of the 20th international conference on World wide web. ACM, 675684

[26] Sejeong Kwon, Meeyoung Cha, and Kyomin Jung. 2017. Rumor Detection over Varying Time Windows. PLOS ONE 12, 1 (2017), e0168344.

[27] Jing Ma, Wei Gao, Zhongyu Wei, Yueming Lu, and Kam-Fai Wong. 2015. Detect rumors using time series of social context information on microblogging websites. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 17511754.

[28] Benjamin Markines, Ciro Cattuto, and Filippo Menczer. 2009. Social spam detection. In Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web. ACM, 4148.

[29] Ke Wu, Song Yang, and Kenny Q Zhu. 2015. False rumors detection on sina weibo by propagation structures. In Data Engineering (ICDE), 2015 IEEE 31st International Conference on. IEEE, 651662.

[30] Zhe Zhao, Paul Resnick, and Qiaozhu Mei. 2015. Enquiring minds: Early detection of rumors in social media from enquiry posts. In Proceedings of the 24th International Conference on World Wide Web. ACM, 13951405.

[31] Sriteja, Allaparthi, Prakhar Pandey, and Vikram Pudi. "Controversy Detection Using Reactions on Social Media." Data Mining Workshops (ICDMW), 2017 IEEE International Conference on. IEEE, 2017.

[32] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473.

[33] Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. arXiv preprint arXiv:1502.03044.

[34] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. arXiv preprint arXiv:1503.08895.

[35] Ankit Kumar, Ozan Irsoy, Jonathan Su, James Bradbury, Robert English, Brian Pierce, Peter Ondruska, Ishaan Gulrajani, and Richard Socher. 2015. Ask me anything: Dynamic memory networks for natural language processing. arXiv preprint arXiv:1506.07285.

[36] Karl Moritz Hermann, Tomas Ko cisk y, Edward Grefen- 'stette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. arXiv preprint arXiv:1506.03340.

[37] Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. 2015. Stacked attention networks for image question answering. arXiv preprint arXiv:1511.02274.

[38] Zhou, Peng, et al. "Attention-based bidirectional long short-term memory networks for relation classification." Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. 2016.

[39] Oriol Vinyals, Lukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever, and Geoffrey Hinton. 2014. Grammar as a foreign language. arXiv preprint arXiv:1412.7449.

[40] Singhania, Sneha, Nigel Fernandez, and Shrisha Rao. "3HAN: A Deep Neural Network for Fake News Detection." International Conference on Neural Information Processing. Springer, Cham, 2017.

[41] Yang, Zichao, et al. "Hierarchical attention networks for document classification." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

[42] Hochreiter, Sepp. "The vanishing gradient problem during learning recurrent neural nets and problem solutions." International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 6.02 (1998): 107-116.

[43] Kyunghyun Cho, Bart Van Merrienboer, Caglar Gul- cehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078

[44] Cho, Kyunghyun, et al. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." arXiv preprint arXiv:1406.1078 (2014).

[45] Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." arXiv preprint arXiv:1409.0473 (2014).

[46] Pennington,J.,Socher,R.,Manning,C.D.:Glove:Globalvectorsforwordrepresentation. In: 19th Conference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 15321543, October 2014

[47] Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP natural language processing toolkit. In: 52nd Annual Meet- ing of the Association for Computational Linguistics (ACL 2014), pp. 5560, June 2014

[48] Kim, Y.: Convolutional neural networks for sentence classification. In: 19th Con- ference on Empirical Methods in Natural Language Processing (EMNLP 2014), pp. 17461751, October 2014

[49] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 14221432.

[50] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 193202. ACM.