Assessing Arabic Weblog Credibility via Deep Co-learning

Chadi Helwe , Shady Elbassuoni , Ayman Al Zaatari and Wassim El-Hajj
Computer Science Department
American University of Beirut
Beirut, Lebanon
{cth05,se58,abz02,we07}@aub.edu.lb

Abstract

Assessing the credibility of online content has garnered a lot of attention lately. We focus on one such type of online content, namely weblogs or blogs for short. Some recent work attempted the task of automatically assessing the credibility of blogs, typically via machine learning. However, in the case of Arabic blogs, there are hardly any datasets available that can be used to train robust machine learning models for this difficult task. To overcome the lack of sufficient training data, we propose *deep co-learning*, a semi-supervised end-to-end deep learning approach to assess the credibility of Arabic blogs. In deep co-learning, multiple weak deep neural network classifiers are trained using a small labeled dataset, and each using a different view of the data. Each one of these classifiers is then used to classify unlabeled data, and its prediction is used to train the other classifiers in a semi-supervised fashion. We evaluate our deep co-learning approach on an Arabic blogs dataset, and we report significant improvements in performance compared to many baselines including fully-supervised deep learning models as well as ensemble models.

1 Introduction

Weblogs, also known as blogs, are gaining popularity, as alternative sources of news and information. The size of the blogosphere is exponentially increasing. For instance, as of October 2018, the popular blogging website Tumblr estimates the total number of blogs on the website to be above 450 million blogs with over 167 billion blog posts[1]. With the surge in misinformation, disinformation and fake news on the Web, and their adverse effects on spreading rumors, tampering with election results and promoting propaganda, an important research question is how to assess the credibility of blog posts. This is particularly crucial in the case of the Arabic speaking world given its recent and constant turmoil.

There has been thus an increased interest in the machine learning and data mining communities to tackle the problem of fake news (Rubin et al., 2016; Wang, 2017; Ruchansky et al., 2017; Zhang et al., 2018; Wang et al., 2018) and the credibility of content in social media in general (Castillo et al., 2011; Gupta and Kumaraguru, 2012; Gupta et al., 2014; El Ballouli et al., 2017; Ma et al., 2016). Some works also focused on the credibility of blog posts (Kolari et al., 2006a,b; Salvetti and Nicolov, 2006; Lin et al., 2007). Most such approaches relied on careful feature-engineering. In this paper, we propose to utilize end-to-end deep learning to assess the credibility of Arabic blog posts. Deep Learning is a type of machine learning that uses deep neural networks to automatically learn features without spending an undue effort to engineer these features as is custom in traditional machine learning. It has been shown to perform significantly better than any other approaches for various NLP tasks. However, deep learning models require a large amount of training data. Assessing the credibility of blog posts is a difficult task and one that has not yet received enough attention from the research community. This has led to only scarce datasets of blogs that are labeled for credibility. This is again particularly true in the case of Arabic blogs, with hardly any such datasets available, with the exception of (Al Zaatari et al., 2016), which only consists of few hundreds of annotated blog posts.

To overcome the lack of sufficient training data, we propose a semi-supervised deep learning approach, which we refer to as *deep co-learning*. Deep co-learning is based on co-training, an approach first introduced by Blum and Mitchell (Blum and Mitchell, 1998) that utilizes multiple

[1] https://www.tumblr.com/about

Proceedings of the Fourth Arabic Natural Language Processing Workshop, pages 130–136
Florence, Italy, August 1, 2019.

classifiers that learn from each other using different views (i.e., features) of the data. In particular, the classifiers are all initially trained in a completely supervised manner using a small training dataset. Each trained classifier is then used to label some unlabeled data, and this automatically labeled data by each classifier is then used to re-train the other classifiers in a semi-supervised fashion.

In our approach, we use a small fully-labeled dataset to train two deep learning models for assessing the credibility of Arabic blog posts. The two classifiers are based on a convolutional neural network (CNN) architecture. The first model uses continuous bag of words (CBOW) word embeddings as features, while the second uses character-level embeddings. We then iteratively retrain our classifiers by applying each classifier on an unlabeled dataset of Arabic blog posts and use the output of each classifier to re-train the other classifier. We evaluate our approach on an Arabic blogs dataset (Al Zaatari et al., 2016) and compare it to various baselines.

Our contributions can be summarized as follows:

- We build an end-to-end deep learning model to assess the credibility of Arabic blog posts
- We utilize semi-supervised learning to train our model even in the lack of sufficient training data
- We evaluate our approach on an Arabic blogs dataset (Al Zaatari et al., 2016) and demonstrate its effectiveness compared to many baselines

The paper is organized as follows. We start by reviewing related work, then describe our deep co-learning approach for assessing the credibility of blog posts. We then present our experimental results where we evaluate our approach on a publicly available Arabic blogs dataset. Finally, we conclude and present future directions.

2 Related Work

Assessing information credibility on the Web is becoming a very hot area of research. Related work that addresses this general problem can be classified into a number of overlapping classes. One such class of works focuses on assessing credibility in social media such as tweets. Another family of works addresses the specific issue of fake news detection. Finally, there are some scarce works on the issue of blog credibility, in which our work also falls.

2.1 Credibility in Social Media

To date, several studies have developed approaches to assess the credibility in Social Media. Castillo et al. (Castillo et al., 2011) implemented automatic methods to predict the level of credibility of a given set of tweets, which was based on various types of features including message-based features, user-based features, topic-based features, and propagation-based features. Gupta and Kumaraguru (Gupta and Kumaraguru, 2012) developed a ranking algorithm to rank tweets, which occurred during high impact events, according to a credibility score. They first identified different features that were used to train a supervised learning model. Their approach is based on a rankSVM model and a relevance feedback method. In a follow-up study, Gupta et al. (Gupta et al., 2014) updated their method to run in a real-time system so that the machine learning model can be retrained from the feedback provided by the user. El Ballouli et al. (El Ballouli et al., 2017) proposed a decision-tree classification model to predict the credibility of Arabic tweets. They extracted different features from tweets and users. Other researches focused on detecting rumors in social media. Ma et al. (Ma et al., 2016) investigated a deep learning approach to detect rumors in microblog platforms such as Twitter and Weibo. They designed a neural network consisting of 2 Gated Recurrent Unit layers that outperformed different baselines.

2.2 Fake News Detection

One of the most important events in 2016 was the U.S presidential election. During this election, fake news began to emerge on social media to sway the votes of electors. Rubin et al. (Rubin et al., 2016) proposed an SVM approach to detect fake news. They used TF-IDF and other features such as absurdity, humor, grammar, negative affect and punctuation. Wang (Wang, 2017) created a benchmark dataset for fake news detection. The dataset consists of 12.8K labeled short political news statements with their meta data. He tested different deep learning models and his best model was a hybrid convolutional and recurrent neural network composed of a convolutional neural network (CNN) trained on the text and another

consisting of a convolutional and a bidirectional long short term memory neural network (CNN-Bi-LSTM) that takes as input the meta data. The outputs of the two models were concatenated and passed to a fully connected layer. Ruchansky et al. (Ruchansky et al., 2017) proposed a hybrid deep learning model to detect fake news. Their model consisted of a recurrent neural network that captures the temporal aspects of articles and a feed-forward fully-connected one that takes as input user features. The output of both neural networks were concatenated and used for classification. Zhang et al. (Zhang et al., 2018) proposed a new deep learning architecture for fake news detection called deep diffusive network. This neural network is based on a gated diffusive unit, which takes as input multiple different sources simultaneously such as news articles, creators and subjects, and then is able to learn to fuse them and output a vector representation that is then used for classification. Finally, Wang et al. (Wang et al., 2018) investigated a deep learning method to detect fake news from newly emerged events.

2.3 Credibility of Weblogs

There is a relatively small body of literature that investigated the assessment of weblogs credibility. Kolari et al. (Kolari et al., 2006a) proposed a machine learning approach to detect spam blogs. They employed a linear support vector machines (SVM) approach that takes as input different features such as TF-Normalized features as well as binary features. Similarly, Salvetti and Nicolov (Salvetti and Nicolov, 2006) implemented a machine learning model to identify spam blogs. They segmented a blog URL into tokens, which were then passed to a Naive Bayes for classification. Lin et al. (Lin et al., 2007) extracted time-based and content-based features that were passed to an SVM classifier. Finally, Al Zaatari et al. (Al Zaatari et al., 2016) constructed a dataset of Arabic blogposts that were labeled for credibility using crowdsourcing. They also manually extracted a handful of features such as bias, sentiment, reasonability and objectivity, and they used these features to train various machine learning models such as Naive Bayes and Decision Tables. However none of these approaches employed end-to-end deep learning as we do in this paper.

3 Deep Co-learning Approach

An overview of our deep co-learning approach is depicted in Figure 1. We use a small fully-labeled dataset to train two deep learning models for assessing the credibility of blog posts. The two classifiers are based on a convolutional neural network (CNN) architecture. The first model uses continuous bag of words (CBOW) word embeddings as features, while the second one uses character-level embeddings. We then iteratively retrain our classifiers by applying each classifier on an unlabeled dataset of blog posts and use the output of each classifier to re-train the other classifier.

In our deep co-learning algorithm (Algorithm 1), we make use of three different datasets. The first dataset D^l, which is a small but *fully-annotated* dataset. This dataset is used to initially train our two CNN models M_1 and M_2 described above. Next, for each one of the two models M_1 and M_2, we pick m random instances from our *unlabaled* dataset D^{ul}. We then apply each of the models M_1 and M_2 on the corresponding m instances we picked for each model.

Next, we iteratively train each of the two co-learning models M_1 and M_2 as follows. We pick k instances out of the m instances on which one of the two models was applied and use them to train the other model. Our goal is to pick the k instances that have the highest accuracy. Once we have computed the score for each instance on which one of the co-learning models were applied, we pick the top-k highest scored instances that were tagged by one model and use it to train the other model and vice versa. Then we use an ensemble averaging of the two models and apply it on our third dataset D^{vl}, which is also a fully-annotated dataset that is used for validation. The validation score of the ensemble average of the two models M_1 and M_2 is stored in the variable $f1\_score$ in each iteration of the deep co-learning algorithm. We check if $f1\_score$ is higher than the current $best\_f1\_score$ and if it is higher, we update the models and augment their datasets with the top-k instances. Then, we set $best\_f1\_score$ to $f1\_score$. Note that the $best\_f1\_score$ is initially set to the validation score of an ensemble averaging of the initial models M_1 and M_2 that were trained using the fully-labeled dataset D^l. We keep repeating this whole process of retrain, apply and pick highest-scored instances for t iterations, which is a hyperparameter in our approach.

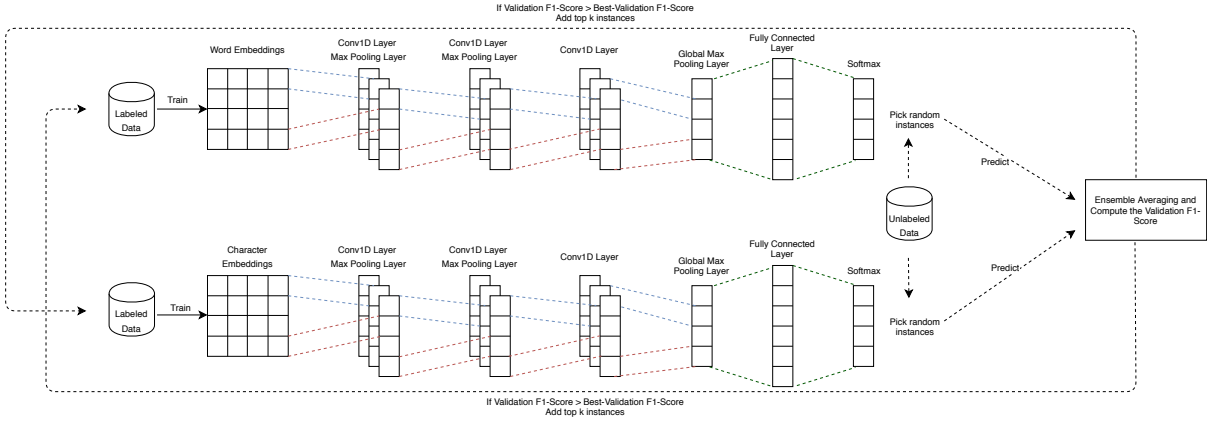

Figure 1: Overview of the Deep Co-Learning Approach

Our approach ends up returning two deep neural network models M_1 and M_2. To be able to use these two models on unseen data, we apply both models and then use ensemble averaging to finally predict the labels of the instances.

In our proposed deep co-learning approach, we utilize two convolutional neural network models. Both of them have the same architecture, except that the first layer of each network utilizes different embeddings. The first model uses pre-trained word-level embeddings that are not retrained in each iteration. However, the second model uses character-level embeddings that are retrained in each iteration. Each model consists of a two 1D convolution layers followed by a max pool layer, and then a 1D convolution layer followed by a global max pool layer. Each convolution layer is composed of 64, 128, and 256 filters, respectively, and a kernel size of 3 and a stride of 1. The max pool layer uses a pool size of 2 and a stride of 2. The output of the global max pool layer is passed to a fully connected layer of 150 neurons. The last layer is a softmax layer of dimension 3. In this architecture, all the hidden layers use RELU as an activation function. In addition, we regularize the neural networks using dropout, and we use a batch normalization layer between all the hidden layers. Figure 2 shows the architecture of the convolutional neural networks used by our deep co-learning approach.

4 Evaluation

To evaluate our deep co-learning approach, we use a dataset of Arabic blog posts constructed by Al Zaatari et al. (Al Zaatari et al., 2016). It consists of 268 Arabic blog posts. The collected blog posts were based on trendy topics at the time of construction, such as Lebanese parliament elections, FIFA world cup, Lebanese residential elections, the Gaza war, the Syrian war, and conflicts in Egypt. To annotate the blogs for credibility, the authors relied on crowdsourcing and the annotators had to label each blog post as credible, fairly credible, or not credible. Note that to the best of our knowledge, this is the only dataset that is publicly available and contains credibility assessment for Arabic blog posts.

We divided the dataset described above as follows: 60% training, 20% validation, and 20% testing. The data was split in a stratified fashion reserving the percentage of samples for each class. Our two deep learning models were bootstrapped using the fully-annotated training dataset, which was used to initially train the co-learning models in the first iteration of the deep co-learning algorithm. We then used the validation dataset to tune the different hyperparameters of our approach. These included the number of instances m we picked at each iteration of the deep co-learning algorithm and the number of instances k with the highest scores. It also included the low-level hyperparameters of the neural networks such as the number of neurons, epochs, and batch size.

In addition to the labeled dataset, we created a large corpus of unlabeled data, which was used to re-train our two deep learning models as described in the previous section. We developed a script to download a set of blog posts from Al Arabiya Blogs [2] and Al Hudood [3]. This dataset consists of 20392 blogs.

We compared our deep co-learning approach to various baselines. The first baseline is a lin-

[2]https://www.alarabiya.net/
[3]https://alhudood.net/

Data: Labeled Data D^l, Unlabeled Data D^{ul}, Validation Data D^{vl}, Iteration t

$D_1^l \leftarrow D^l$
$D_2^l \leftarrow D^l$
$M_1 \leftarrow train(D_1^l, WordLevelEmbeddings)$
$M_2 \leftarrow train(D_2^l, CharLevelEmbeddings)$
$best\_f1\_score \leftarrow Avg(M_1, M_2, D^{vl})$
repeat
  $D_1^{sl} \leftarrow$ Pick m random instances from D^{ul}
  $D_2^{sl} \leftarrow$ Pick m random instances from D^{ul}
  Apply $(M_1, D_1^{ul}, CBOW)$
  Apply $(M_2, D_2^{ul}, Skip-gram)$
  for *i = 1 to m* **do**
    Compute s_i for each instance $i \in D_1^{ul}$
    Compute s_i for each instance $i \in D_2^{ul}$
  end
  $TmpD_1^l \leftarrow D_1^l \cap top\text{-}k_2$
  $TmpD_2^l \leftarrow D_2^l \cap top\text{-}k_1$
  $TmpM_1 \leftarrow train(TmpD_1^l, WordLevelEmbeddings)$
  $TmpM_2 \leftarrow train(TmpD_2^l, CharLevelEmbeddings)$
  $f1\_score \leftarrow Avg(TmpM_1^l, TmpM_2^l, D^{vl})$
  if $f1\_score > best\_f1\_score$ **then**
    $top-k_1 \leftarrow$ Remove top-k instances with highest s_i from D_1^{ul}
    $top-k_2 \leftarrow$ Remove top-k instances with highest s_i from D_2^{ul}
    $D_1^l \leftarrow TmpD_1^l$
    $D_2^l \leftarrow TmpD_2^l$
    $M_1 \leftarrow TmpM_1$
    $M_2 \leftarrow TmpM_2$
    $best\_f1\_score \leftarrow f1\_score$
  end
until t *iterations*;
return M_1, M_2

Algorithm 1: Deep Co-learning Algorithm

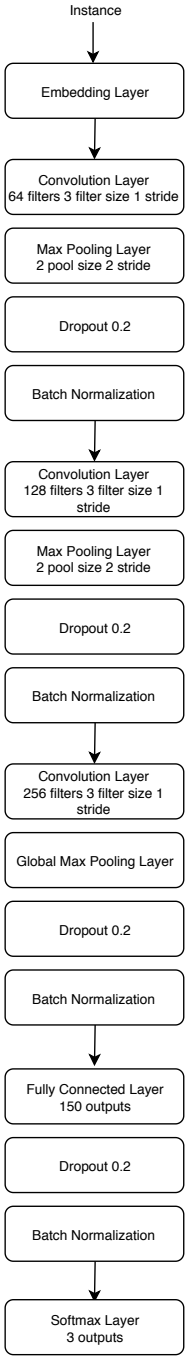

Figure 2: Convolutional Neural Network Architecture

ear SVM that is trained using the TF-IDF scores of the words in the blog posts, and we set the soft-margin weight C to 5 based on the validation set. This baseline is used to evaluate the effectiveness of a deep-learning approach such as ours compared to a more simple model such as SVM. The second and third baselines are word-level convolution neural networks (Word-CNN), and a character-level convolution neural networks (Char-CNN), respectively. The last baseline we compared our deep co-learning approach to is an ensemble model of Word-CNN and Char-CNN (Ensemble CNN). All the model were trained on the same training dataset, and their hyperparemeters were tuned using the same validation set.

We trained all supervised models (i.e., the first two baselines and the initial models of the deep co-learning approaches) for 500 epochs with a batch size of 16, a dropout of 0.2 after each hidden layer, and we used Adagrad (Duchi et al., 2011) as

Model	F1-Score
SVM TF-IDF	0.57
Word-CNN	0.52
Char-CNN	0.54
Ensemble CNN	0.50
Deep Co-learning	**0.63**

Table 1: Evaluation Results

the optimization algorithm. All experiments were run on a Ubuntu machine with a 24 GB RAM, a CPU Intel Core I7 and a GPU NVIDIA GeForce GTX 1080 Ti 11GB. For the deep co-learning approaches, we repeated the process of co-learning for 50 times since retraining the models was taking significant time which is around 24 hours. In each iteration of the co-learning algorithm, we randomly picked 1000 sentences from the unlabeled data and used the top-50 scored sentences to retrain the other model. All the other parameters were adjusted using the validation set. Note that we also experimented with variations of the above, but we only report here the best performing ones based on validation data.

Table 1 shows the results of our deep co-learning approach and the baselines on the testing dataset. We observe that an SVM model trained with TF-IDF scores as features has an F1-score of 0.57, which is higher than all the fully supervised deep learning approaches. This can be mainly attributed to the small size of the training dataset, which makes it harder to train more complex models such as the fully-supervised deep learning models. Comparing the fully-supervised deep learning models to each other, we observe that the deep learning model trained on character-level representations has an F1-Score of 0.54, while the deep learning model trained on word-level representations has a lower F1-score of 0.52. The advantage of character-level models over word-level models is that they can learn misspellings, emoticons, and n-grams. Interestingly, the ensemble model of Word-CNN and Char-CNN (Ensemble CNN in Table 1) performs worse than all other models. This indicates that with the lack of enough training data, even ensemble models are not able of generalizing well. On the contrary, our deep co-learning approach, which combines the best of both worlds, the complexity of deep learning approaches and the ability to generalize well even when no sufficient training data is available through semi-supervision, significantly outperforms all the baselines with an F1-Measure of 0.63.

5 Conclusion and Future Work

In this paper, we proposed a deep learning approach to assess the credibility of Arabic blog posts. Our method, deep co-learning, is based on a semi-supervised learning algorithm known as co-training that we adopted to the realm of deep learning. To train our deep co-learning approach, we generated an unlabeled dataset that was then used to train our deep co-learning approach. We evaluated our approach on an Arabic blogs dataset and compared it to different baselines. Our deep co-learning approach significantly outperformed all other compared-to approaches including both deep and traditional machine learning models.

In future work, we plan to train the deep co-learning approach for a more extended period to improve its performance. We also plan to label some of our unlabelled blog posts that we used for training our deep co-learning approach using crowdsourcing and to make the labeled dataset publicly available to advance research in this area. Finally, we also plan to experiment with other neural network architectures and to incorporate more linguistic features in our models.

References

Ayman Al Zaatari, Rim El Ballouli, Shady Elbassuoni, Wassim El-Hajj, Hazem M. Hajj, Khaled B Shaban, Nizar Habash, and Emad Yahya. 2016. Arabic corpora for credibility analysis. In *LREC*.

Avrim Blum and Tom Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100. ACM.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th international conference on World wide web*, pages 675–684. ACM.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Rim El Ballouli, Wassim El-Hajj, Ahmad Ghandour, Shady Elbassuoni, Hazem Hajj, and Khaled Shaban. 2017. Cat: Credibility analysis of arabic content on twitter. In *Proceedings of the Third Arabic Natural Language Processing Workshop*, pages 62–71.

Aditi Gupta and Ponnurangam Kumaraguru. 2012. Credibility ranking of tweets during high impact events. In *Proceedings of the 1st workshop on privacy and security in online social media*, page 2. ACM.

Aditi Gupta, Ponnurangam Kumaraguru, Carlos Castillo, and Patrick Meier. 2014. Tweetcred: Real-time credibility assessment of content on twitter. In *International Conference on Social Informatics*, pages 228–243. Springer.

Pranam Kolari, Tim Finin, Anupam Joshi, et al. 2006a. Svms for the blogosphere: Blog identification and splog detection. In *AAAI spring symposium on computational approaches to analysing weblogs*.

Pranam Kolari, Akshay Java, Tim Finin, Tim Oates, Anupam Joshi, et al. 2006b. Detecting spam blogs: A machine learning approach. In *Proceedings of the national conference on artificial intelligence*, volume 21, page 1351. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Yu-Ru Lin, Hari Sundaram, Yun Chi, Junichi Tatemura, and Belle L Tseng. 2007. Splog detection using self-similarity analysis on blog temporal dynamics. In *Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, pages 1–8. ACM.

Jing Ma, Wei Gao, Prasenjit Mitra, Sejeong Kwon, Bernard J Jansen, Kam-Fai Wong, and Meeyoung Cha. 2016. Detecting rumors from microblogs with recurrent neural networks. In *IJCAI*, pages 3818–3824.

Victoria Rubin, Niall Conroy, Yimin Chen, and Sarah Cornwell. 2016. Fake news or truth? using satirical cues to detect potentially misleading news. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*, pages 7–17.

Natali Ruchansky, Sungyong Seo, and Yan Liu. 2017. Csi: A hybrid deep model for fake news detection. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 797–806. ACM.

Franco Salvetti and Nicolas Nicolov. 2006. Weblog classification for fast splog filtering: A url language model segmentation approach. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 137–140. Association for Computational Linguistics.

William Yang Wang. 2017. " liar, liar pants on fire": A new benchmark dataset for fake news detection. *arXiv preprint arXiv:1705.00648*.

Yaqing Wang, Fenglong Ma, Zhiwei Jin, Ye Yuan, Guangxu Xun, Kishlay Jha, Lu Su, and Jing Gao. 2018. Eann: Event adversarial neural networks for multi-modal fake news detection. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 849–857. ACM.

Jiawei Zhang, Limeng Cui, Yanjie Fu, and Fisher B Gouza. 2018. Fake news detection with deep diffusive network model. *arXiv preprint arXiv:1805.08751*.