All are not liars:
Fine-grained fact-checking in political statements

Yixuan Meng, Andrew Song, Haoran Liu
School of Engineering and Applied Science, University of Pennsylvania
{yixuanm, songdrew, haoranl}@seas.upenn.edu

Abstract

The purpose of this paper is to improve current fake news detection algorithms using the LIAR dataset. Many models use a text only based approach in detection; we present and analyze the differences in performance between text and hybrid models that incorporate both text and metadata. We implement a CNN model as the text only model and experiment with state-of-the-art tools (Word2Vec, GloVe, XLNet) to build our hybrid models. The average accuracy of published baselines reach around 21%; our best hybrid model achieves an accuracy of 27.94%. Through the paper, we show the strengths of using different technologies in building neural networks and methods such as early concatenation.

1 Introduction

With the low cost, easy access, and rapid dissemination of information, people seek out and consume news from social media and web sources. But these exact sources enable the spread of low quality news with intentions to spread false information. The extensive spread of fake news has the potential for extremely negative impacts on individuals and society, and through this paper we are excited to tackle this problem. Natural language processing is at the heart of fake news detection: we must use the text and other information given in order to make the judgment that the text has fallacious or correct information. Either through traditional machine learning models such as SVM or naive bayes to recent advancements in neural networks, fake news detection has largely been classified as a NLP task due to its relevance to text.

As shown in Figure 1, the formal definition of our task is: given text (with or without metadata), classify the truthfulness of the text into six categories: pants-fire, false, barely-true, half-true, mostly-true, and true. The reason for these classes are explained in the literature review citing Wang (2017).

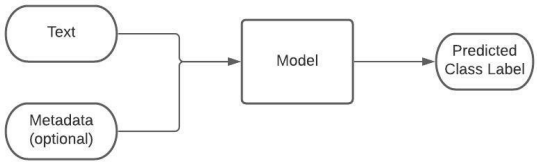

Figure 1: Classfication Task

We picked this task for our final project because our nation has recently undergone many misfortunes due to fake news, such as new surrounding the efficacy of COVID-19 vaccines. Further, previous datasets only allows text-only models (Mitra and Gilbert, 2015), and we were interested in worked with metadata, multi-label transfer learning (Tao and Fang, 2020; Banerjee et al., 2019), and state-of-the-art technologies such as XLNet (Yang et al., 2019a). We also learn the importance of different evaluation metrics (Macro F1 being much more relevant to our task). In all, we were interested in getting our hands dirty with popular neural network architectures, word embeddings, and models used today.

The key components of this project are summarized as follows:

- We complete a majority vote baseline and use a text-only CNN model as a strong baseline.
- We add Macro F1 score to our evaluation instead of only using accuracy.
- We implement a BiLSTM with attention model and compare the performance of early and late fusion models, and the performance of the GloVe and the Word2Vec embedding.
- We integrate metadata from the Liar and Liar-Plus dataset to our models.

- We transfer the pretrained XLNet model to this dataset.
- Our BiLSTM with attention model and the transfer learning based approach reaches state-of-the-art performance.

2 Literature Review

Wang (2017) proposed the LIAR dataset for fake news detection. The task is to classify a statement into six categories: pants-fire, false, barely-true, half-true, mostly-true, and true. To build a better fake news detector, the author proposed a hybrid CNN model which integrates the meta-data and the text. In the model, the embedded meta-data are feed to a conventional layer, then the output of it is sent to a Bi-LSTM layer. The output of the Bi-LSTM layer is concatenated with the output of the max-pooling layer of a text-only CNN model. Then the concatenated vector is feed to a fully connected layer with a softmax predictor to get the prediction.

In the experiment, each meta-data are used separately to compare the performance of hybrid CNN models, the CNN with text and speaker meta-data has the highest accuracy (0.277) on the valid dataset, and the CNN integrating text and all meta-data has the highest accuracy (0.274) on the test dataset.

Shu et al. (2017) present an overview of the current methods used to detect fake news on social media. The authors proposed a general data mining framework for this task with two phases: feature extraction and model construction. The goal of feature extraction is to encapsulate news content and related information in a formal mathematical model. Models based on news content features can be split into two categories: knowledge-based and style-based. And social context models can be classified into two categories: stance-based and propagation-based.

Shu et al. (2017) points out that news content could not provide enough information for fake news detection. Fake news is written to mislead news consumers on purpose, which makes it not satisfactory to detect fake news based on its content. Thus, building comprehensive datasets is necessary and helpful for fake news propagation, detection, and mitigation. The author of (Shu et al., 2020) provide a fake news data repository *FakeNewNet*, which contains two comprehensive datasets that includes *news content*, *social context*, and *dynamic information*.

Social context and dynamic information facilitate the accuracy of deciding whether the news is fake by including users' engagements and social behaviors in our model. Social context includes users' behaviors towards each news such as replies, likes, and reposts. Also, user profiles, user posts, and social network information are worthwhile to be collected. Meanwhile, dynamic information measures how fake news propagate on social media, and how the changing of fake news' topics over time.

Finally, the author of (Shu et al., 2020) tested the performance of several state-of-the-art baseline models on fake news detection using FakeNewsNet. The results turn out FakeNewsNet has the potential to facilitate many research directions such as fake new detection, mitigation, evolution, malicious account detection, etc.(Shu et al., 2020)

3 Experimental Design

3.1 Dataset

We use the LIAR dataset in our project. This dataset is collected from POLITIFACT.COM's API[1] (Wang, 2017), which includes 12.8 thousand short statements that are labeled into six categories: pants-fire, false, barely-true, half-true, mostly-true, and true. A strong reason why we found this dataset interesting and promising is because LIAR provides meta-data beyond the text such as 'speaker' and 'job_title'. Specifically, the features of this dataset along with an example are:

[id, label, statement, subject, speaker, job_title, state_info, party_affiliation, barely_true_counts, false_counts, half_true_counts, mostly_true_counts, pants_on_fire_counts, 'context']

[2635.json, false, 'Says the Annies List political group supports third-trimester abortions on demand.', abortion, dwayne-bohac, State representative, Texas, republican, 0, 1, 0, 0, 0, a mailer]

Further, we provide summary statistics of the dataset:

The distribution of the six labels in the training dataset are shown in Figure 2, ordered by the severity of its falseness, from 'true' to 'pants-fire' (which means the statement is totally false).

The metadata 'party affiliation' allows us to answer a question: politicians from which parties lies more? We compared the percentiles of each label

[1] http://static.politifact.com/api/ v2apidoc.html

Summary Statistics	
Training Set Size	10,269
Validation Set Size	1,284
Test Set Size	1,283
Avg. Statement Length	17.9
Top 3 Speaker Affiliations	
Democrats	4,150
Republicans	5,687
None (eg FB posts)	2,185
Distinct Counts (Training)	
Speakers	2911
Contexts	4346
Subjects	3828

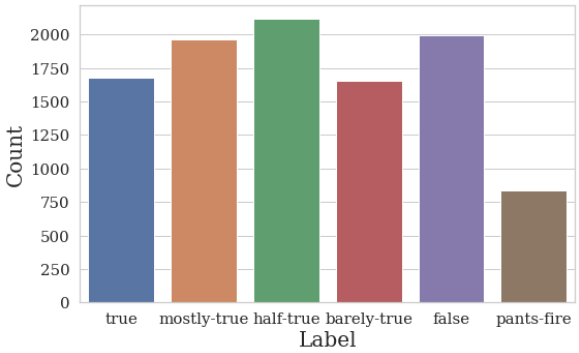

Figure 2: Distribution of six labels

for the biggest two parties, the Democratic Party and the Republican Party. The frequency of appearance of each label in the republicans' statements increases from left to right in Figure 4, and a significantly higher rate of 'pants on fire' events is spotted for the party.

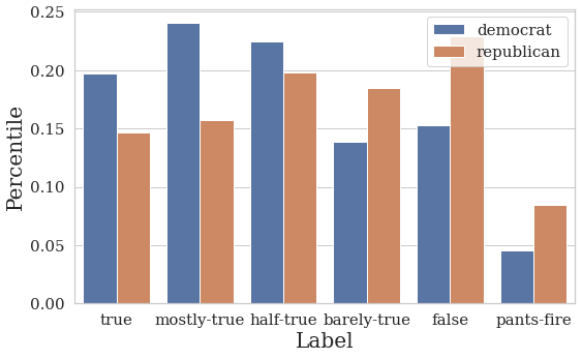

Figure 3: Comparison of lies in different parties

Similarly, we take a look into the influence of another metadata 'job title', and select three most frequent job titles in the dataset, which are 'President', 'U.S. Senator', and 'Governor'. Job title 'President-Elect' and 'Presidential candidate' are also counted into 'President' in this case. As shown in Figure 4, senators lies slightly less according to the 75 percentile, the presidents went through more 'pants on fire' events, but also speaks more truth than the governors.

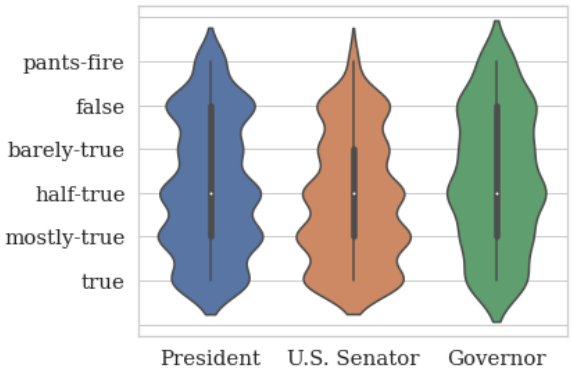

Figure 4: Distribution of labels for people with most frequent job titles

Overall, we believe that this dataset will provide us with the necessary features (text and context) to add our extensions to fo the fake-checking.

3.2 Evaluation Metrics

We use two main evaluation metrics for the task based on literature. The first is just accuracy and the other is macro F1, which is computing the f1 score across multiple labels.

In plain English, the accuracy of a model is the number of times that the model guessed the correct class correctly. For example, if our model classified a pants-fire text as true, then the accuracy will lower. Formally, the accuracy is $\frac{\sum_{n=1}^{n} \mathbf{1}\{\text{Class Label} = \text{Prediction Label}\}}{\sum_{n=1}^{n} 1}$. García et al. (2009)

Macro F1, in plain English, is the F1 score weighted across different labels. The F1 score is a metric that determines how well a model did based on how accurately it predicted on its guesses, as well as on just correct labels. Formally, the Macro F1 is: $\frac{1}{n}\sum_{i=1}^{n} \text{F1-score}_i$, and the F1 score is defined as $2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$. Opitz and Burst (2021)

Examples of citations that used the accuracy and macro f1 metrics are Wang (2017), Ghanem et al. (2018), and Reis et al. (2019)

3.3 Simple baseline

The simple baseline that we implement is a majority classifier. Therefore, this classifier naively predicts the class that was seen most often in the

training set. This file takes in as input the training set and the test set. The test set does not need to have labels. The output is a csv file in a single line that contains all of the predictions, which will all be the same class. For convenience, the file also outputs the test labels if given. For example, the output (predictions) of a test dataset that contains 5 instances would be:

half-true, half-true, half-true, half-true

The score of this simple majority classifier on our test set using out evaluation script is 20.92%. Also, the majority vote baseline reaches up to 19.31% accuracy by predicting all statements to be 'false'. Interestingly, there are six classes that our model must predict from, which means that there is a slight class imbalance in the test dataset. Nonetheless, the accuracy is, as predicted, not high and is just a simple baseline.

4 Experimental Results

4.1 Published baseline

We implement a CNN model as the strong baseline for the liar dataset. The liar dataset is loaded from huggingface datasets, we use the pre-trained word2vec embedding with 300 dimensions to generate embedding for each statement in the dataset. Each statement are preprocessed by removing special characters in them and converting all words to lower case. The predefined train, validation, and test dataset are shuffled and loaded with 64 batch size for the training.

We tune the parameter of the CNN model on the validation dataset. There are three convolution layers in the model, the filter size are 5,5,5 separately. Each size has 128 filters. The dropout rate is set to 0.8. We concatenate the output of the three convolution layers and feed it to fully connected layer to do the six-way classification. For the training, we use the Adam optimizer, set the learning rate to 0.001, use cross entropy loss and train for 10 epochs. The accuracy of this CNN model is 0.2368 on the validation dataset, and 0.2182 on the test dataset.

The result of this strong baseline, and the performance of another two common models, a logistic regression classifier and a support vector machine classifier are shown in Table 1. The published baseline (Wang, 2017) used a CNN model with the same structure, and its the accuracy on the validation set was 26%, which is higher than our model's 23.68%. The CNN in the published paper gives best performance when the kernel size is set to 2,3,4, and the dropout set to 0.8, which is slightly different from our optimal parameters. It may caused by different ways of preprocessing the texts, and the random order of our data loader.

4.2 Extensions

4.2.1 Extension 1

For the first extension, we change the model structure and implement a BiLSTM with attention model. We also add a new metadata column 'justification' from the Liar-Plus dataset(Alhindi et al., 2018) and integrate all metadata into our BiLSTM model.

The BiLSTM with attention model inputs the word embedding to BiLSTM layers, and feed the output of the BiLSTM layers to an attention layer. A fully connected layer is added in the end to do the six-way classification. After changing the model structure from CNN to BiLSTM, the accuracy on the validation set is improved from 23.68% to 26.71%.

In the strong baseline model, we only examine the text data by generate word embeddings. With metadata, we are able to incorporate more features such as political party, who the text was written from, and the subject of the text. Specifically, the following columns of metadata are used

'subject', 'speaker', 'job_title' 'state_info'

'party_affiliation','context','justification'

When tokenizing, a separator is added between each metadata, for instance, for metadata 'speaker' and 'job_title', the generated tokens are:

barack-obama + [SS] + *President*

We initially use the Word2Vec embeddings for both the statement text and the metadata. Then we find that GloVe embedding is a better option for the metadata, and we use the same embeddings (Word2Vec) for the text. We set the hidden dimension of the BiLSTM layers to be 48, and number of layers to be 2, the dropout rate is optimized to 0.3. We compared the performance of this model when concatenating the embedded metadata vector and the statement vector before feed them into the BiLSTM layers (early fusion), and use a separate BiLSTM layers to extract features from embedded

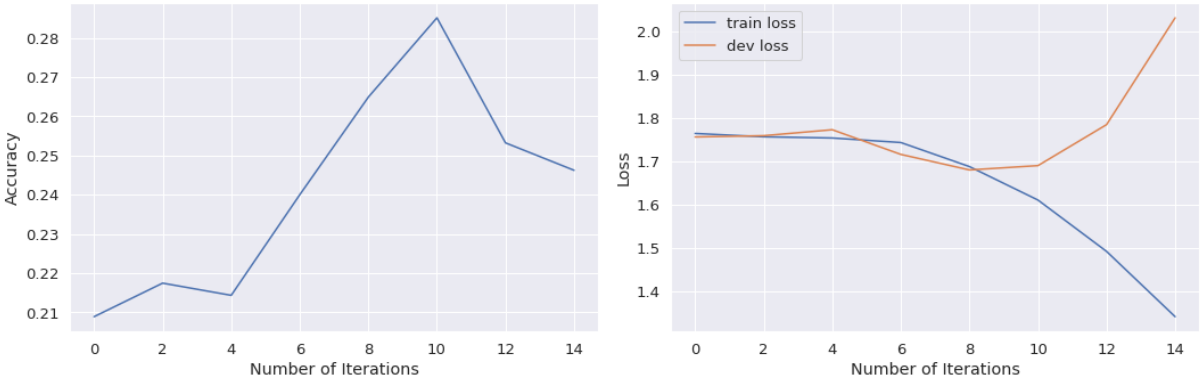

Figure 5: Plot of validation, training loss, and validation accuracy during training for best BiLSTM with attention model.

metadata and concatenate the output of attention layer and the BiLSTM layers' later (late fusion). As shown in Table 1, though according to Shu et al. (2017), late fusion improves the performance of the author's hybrid CNN model, in our experiment, the early fusion gives a much higher accuracy on both the validation and test dataset, indicating that using separate BiLSTM layers for feature extraction sabotages the performance.

To sum up, we use the GloVe embedding for the metadata, and the Word2Vec embedding for the statement text, and train the BiLSTM with attention model with batch size 64, learning rate 0.0002 for 10 epochs, we get the best performance on the test set which is 27.62% accuracy, the loss and accuracy curves are shown in Figure 5. Increasing the batch size to 128 seems to introducing overfitting, because it increases the difference between validationa and test accuracy by a large margin, as shown in Table 1.

4.2.2 Extension 2

Transfer learning uses new data to fine tune the pretrained models. It is much less expensive than training from scratch. We use transfer learning method to test the ability of generalization of pretrained model to the fact-checking task on this dataset.

We incorporates the XLNet-base-cased pretrained model from HuggingFace. Yang et al. (2019b) XLNet is an unsupervised language representation method similar to BERT, but uses permutation language modeling to learn from more contexts.

In order to incorporate the model, we import the XLNetTokenizer and XLNetModel from the transformers package also in HuggingFace. We use AdamW optimizer for our model because of its benefits in adapting to different paramaeters. For learning, we use a cosine schedule using warm-up since we believed that having a cycle of large learning rates to decreasing learning rates to large again would be beneficial in learning different scenarios from medatada and text. We padded the text vector to the length 28, which is the 90 percentile of the length of all texts. Similarly, the metadata vector is padded to the length 132.

We add a fully connected layer and only update the parameters of the last four layers and freeze all other layers. (Tao and Fang, 2020) Specifically, our model had a learning rate of .00015. batch size of 16, and ran for 15 epochs.. These numbers were found using hyperparameter tuning, 6 shows the effect of running over 15 iterations. Running for longer decreased our accuracy due to overfitting. This model achieved our best accuracy of 27.94%.

4.3 Error analysis

We evaluate the XLNet model because it was our most successful model and has the highest test accuracy. The confusion matrix (of only errors) is shown in Figure 7. Our model predicts pants-fire statements with most accuracy, and mostly-true statements with least accuracy. Other categories that our model struggles on are true and mostly true statements, and other categories that our model does well on are false statements. Interestingly, our model does better on statements that veer towards false, and has a harder time for predicting truer statements. We believe this is because false statements follow similar patterns of yellow journalism and flagrancy. Our extensions did better than the published baseline, and an example of a correct guess is the following : "We know that more than half of Hillary Clintons meetings while she was

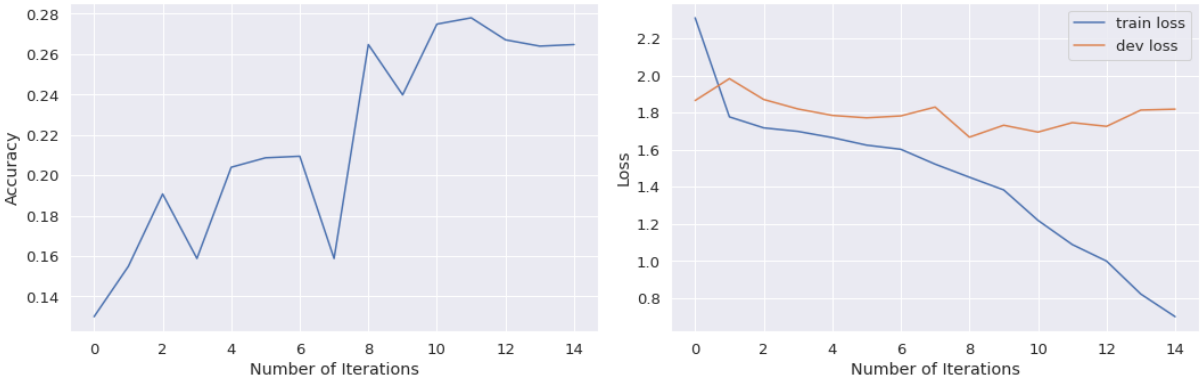

Figure 6: Plot of validation, training loss, and validation accuracy during training for best XLNet.

Models	Valid Acc	Valid Macro F1	Test Acc	Test Macro F1
Majority	19.31	5.40	20.92	5.77
Text-only CNN	23.68	11.35	21.82	10.21
Logistic Regression	19.63	13.32	21.43	14.93
SVM	22.74	15.46	19.95	13.32
Hybrid BiLSTM+Attention (Early fusion, 64 batch)	**28.50**	**24.18**	**27.62**	**23.39**
Hybrid BiLSTM+Attention (Early fusion, 128 batch)	29.52	23.69	26.36	20.60
Hybrid BiLSTM+Attention (Late fusion)	19.08	16.56	18.39	15.94
Hybrid BiLSTM+Attention (Early fusion, no GloVe)	26.64	20.52	25.34	18.75
Hybrid XLNet (Early fusion, 15 epochs)	**27.80**	**24.02**	**27.94**	**23.56**
Hybrid XLNet (Early fusion, 20 epochs)	28.04	27.92	24.94	23.22

Table 1: The evaluation results of our models (unit: %).

secretary of state were given to major contributors to the Clinton Foundation." which has a true label of "barely-true." Our XLNet correctly guesses this example opposed to our CNN model, which guessed this statement has being "false." Though they both predict that they are in the realm of fake news, XLNet is more accurate in saying that the statement is barely-true rather than outright false.

5 Conclusions

Both of our extensions reached state-of-the-art performance[2]. The model proposed in Wang (2017) have an average accuracy around .25 accuracy. Our Hybrid BiLSTM + Attention model that used early fusion of text and metadata achieved an accuracy of .2762 and Macro F1 of .2339 on the test set. Further, our Hybrid XLNet model that also used early fusion of text and metadata achieved an accuracy of .278 and Macro F1 of 23.56.

We are pleased with our results, and the potential implications of our project. Interestingly, as stated in Extension 1, we discovered that early fusion of text embeddings and metadata embeddings achieved better accuracies than late fusion, which is recommended by Shu et al. (2017). We believe that this makes more sense since our models will learn the text along with its context much earlier. Perhaps this reason may be due to the structure of our network and its differences

[2]https://paperswithcode.com/sota/fake-news-detection-on-liar

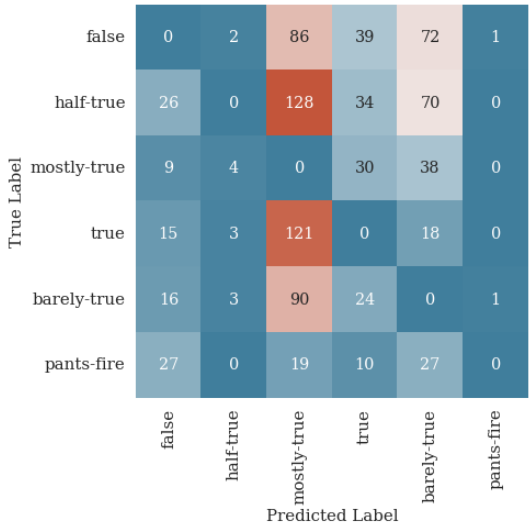

Figure 7: XLNet model confusion matrix

with those mentioned in the paper.

We are also curious about the moral concerns of using metadata, such as political affiliation. Our models used this information to achieve a jump in accuracy, but this may discriminate against some groups as producing more fake news. In all, we are pleased with the results and experience.

6 Acknowledgements

Thank you to our TA Siyi! He was super receptive and helpful :)

References

Tariq Alhindi, Savvas Petridis, and Smaranda Muresan. 2018. Where is your evidence: improving fact-checking by justification modeling. In *Proceedings of the first workshop on fact extraction and verification (FEVER)*, pages 85–90.

Siddhartha Banerjee, Cem Akkaya, Francisco Perez-Sorrosal, and Kostas Tsioutsiouliklis. 2019. Hierarchical transfer learning for multi-label text classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6295–6300.

Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. 2009. A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. *Soft Computing*, 13(10):959.

Bilal Ghanem, Paolo Rosso, and Francisco Rangel. 2018. Stance detection in fake news a combined feature representation. In *Proceedings of the first workshop on fact extraction and VERification (FEVER)*, pages 66–71.

Tanushree Mitra and Eric Gilbert. 2015. Credbank: A large-scale social media corpus with associated credibility annotations. In *ICWSM*.

Juri Opitz and Sebastian Burst. 2021. Macro f1 and macro f1.

Julio CS Reis, André Correia, Fabrício Murai, Adriano Veloso, and Fabrício Benevenuto. 2019. Supervised learning for fake news detection. *IEEE Intelligent Systems*, 34(2):76–81.

Kai Shu, Deepak Mahudeswaran, Suhang Wang, Dongwon Lee, and Huan Liu. 2020. Fakenewsnet: A data repository with news content, social context, and spatiotemporal information for studying fake news on social media. *Big data*, 8(3):171–188.

Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu. 2017. Fake news detection on social media: A data mining perspective. *ACM SIGKDD explorations newsletter*, 19(1):22–36.

Jie Tao and Xing Fang. 2020. Toward multi-label sentiment analysis: a transfer learning based approach. *Journal of Big Data*, 7(1):1–26.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019a. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime G. Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. *CoRR*, abs/1906.08237.