SkoltechNLP at SemEval-2020 Task 11: Exploring Unsupervised Text Augmentation for Propaganda Detection

Daryna Dementieva, Igor Markov, and Alexander Panchenko
Skolkovo Institute of Science and Technology
Bol'shoy Bul'var, 30, Skolkovo Innovation Center, Moscow Oblast, 143026, Russia
{Daryna.Dementieva, Igor.Markov, A.Panchehko}@skoltech.ru

Abstract

This paper presents a solution for the Span Identification (SI) task in the "Detection of Propaganda Techniques in News Articles" competition at SemEval-2020. The goal of the SI task is to identify specific fragments of each article which contain the use of at least one propaganda technique. This is a binary sequence tagging task. We tested several approaches finally selecting a fine-tuned BERT model as our baseline model. Our main contribution is an investigation of several unsupervised data augmentation techniques based on distributional semantics expanding the original small training dataset as applied to this BERT-based sequence tagger. We explore various expansion strategies and show that they can substantially shift the balance between precision and recall, while maintaining comparable levels of the F1 score.

1 Introduction

Propaganda is one of the primary indicators of false news. Therefore, the task of detecting and classifying propaganda is an important one in the field of fake news detection. Da San Martino et al. (2019) presented a new dataset for propaganda detection. The main advantage of this dataset is the markup at the level of individual text fragments. In addition to binary markup (is/is not propaganda), there is also a multiclass markup, which includes 18 different propaganda classes. Using this dataset, the Span Identification (SI) task of Detection of Propaganda Techniques in News Articles task requires to determine specific text fragments which contain at least one propaganda technique. For instance, in the sentence *"Manchin says Democrats acted like babies at the SOTU (video) Personal Liberty PollExercise your right to vote"* the part from the 34-th through the 40-th character (i.e., word *"babies"*) belongs to the Name-calling and Labeling class, so it should be marked as propaganda.

This paper describes the solution by "SkoltechNLP" team which took part in the SI task of the competition and achieved a score $F1 = 0.34$ on the test set evaluation. Our solution is based on BERT (Devlin et al., 2019) that is specially pretrained for the sequence tagging task. However, such architectures usually require large datasets for fine-tuning: for instance, CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) consists of 976 articles in the training set, in comparison with 371 articles in the training set for the propaganda detection task. Therefore, we conducted a study of the dataset expansion with several augmentation techniques. We built a number of strategies, performing investigation of various combinations of such parameters as: the size of the enlarged dataset, models for word representations, words' part of speech, and class for substitution. All codes for reproducing the results are openly available online.[1]

The rest of the paper is organized as follows. Section 2 covers several previous approaches to this problem. Section 3.1 presents our final solution and describes model architecture used, preprocessing, and implementation details. Section 4 provides the description and the results of the dataset expansion experiments. Finally, Section 5 concludes this report.

[1] https://github.com/skoltech-nlp/semeval2020-propaganda

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 1786–1792
Barcelona, Spain (Online), December 12, 2020.

2 Related Work

Yoosuf and Yang (2019) proposed a solution for Fragment Level Classification (FLC) task in the Fine Grained Propaganda Detection competition at the NLP4IF'19 workshop. The participants had a similar task as in "Detection of Propaganda Techniques in News Articles" competition of SemEval 2020 to detect text fragments with propaganda. The difference is that the markup was at the level of whole sentences. As a result, the authors solved the problem of determining the sentence to one of the 19 classes (without propaganda or one of the 18 types of propaganda). To solve this problem they used model based on BERT Language Model with linear classification head for token classification. They also tried several techniques to overcome the lack of data and classes imbalance: 1) weighting rarer classes with higher probability; 2) sample propaganda sentences with a higher probability than non-propaganda sentences.

Ek and Ghanimifard (2019) describe solution also for the same competition at the NLP4IF'19 workshop. As a classification model they use BiLSTM. In addition to the model development, the authors investigate different augmentation techniques for balancing classes. They used synthetic-minority over-sampling (Chawla et al., 2002) algorithm to generate token embeddnings for the minority classes in the dataset. They used three models for contextual embeddings – ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) and GROVER (Zellers et al., 2019). Out of these models, ELMo showed the overall best F1-score for classes in the FLC task. However, for individual classes, the best model varies.

Since in the previous competition the participant with successful solutions focused more on pre-trained contextualized models, we also decided in our approach to focus on such models, BERT in particular. Moreover, as data augmentation applications were used in previous works in propaganda detection (Ek and Ghanimifard, 2019) and also have shown significant results in other fields like computer vision (Krizhevsky et al., 2012), it seems promising to continue research in this direction.

3 Method

We approach the problem as Named Entity Recognition (NER) problem with two classes – inside and outside of propaganda. Since models for such a task usually use token classification and the markup was made on a character level, firstly, we made a preprocessing step that converts char-level markup into token-level markup. At the post-processing step, we did the reverse transformation of the markup. The pipeline of our final solution is presented in Figure 1.

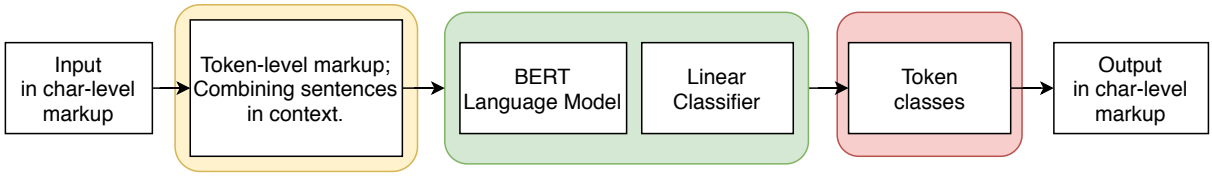

Figure 1: The pipeline of our final solution of the Span Identification (SI) task (**BERT-Linear**).

3.1 Model

During the competition, we conducted experiments with different models. The first one was **BiLSTM-CNN-CRF** model (Ma and Hovy, 2016) as implemented by Chernodub et al. (2019). This is a commonly used approach for NER and sequence labelling task: it uses both word-level and char-level embeddings that are fed to BiLSTM-CNN-CRF module. The second one, denoted as **BERT-Linear**, relies on a linear classifier on the top of BERT-based token representations. The implementation of this sequence tagger is based on BertForTokenClassification class from the transformers library[2] as done by Shelmanov et al. (2019). Finally, we also tried **BERT-CRF** model[3]: after BERT classifier a Viterbi decoder is used for better tags sequence approximation. Our final submission is based on the second model architecture as it yielded overall better results, as described below.

[2] https://github.com/skoltech-nlp/transformers_sequence_tagger
[3] https://github.com/Louis-udm/NER-BERT-CRF

	dev			test		
	F1	Precision	Recall	F1	Precision	Recall
Baseline	0.0110	0.1100	0.0058	0.0030	0.1304	0.0015
BiLSTM-CNN-CRF	0.0561	0.1093	0.0377	-	-	-
BERT-CRF	0.3547	**0.3519**	0.3630	-	-	-
BERT-Linear	**0.3670**	0.3498	**0.3858**	0.3406	0.4652	0.2687

Table 1: Our submission results on the SI task on the public (dev) and private (test) leaderboards.

3.2 Preprocessing

Our solution performs token-level classification, while the data labels are at the character level. Standard libraries for tokenization did not work for us, as it was noticed that during the reverse transformation from token-level to character level markup index shift occurred. Therefore, we developed our method for conversion to the correct markup. We can distinguish such features of preprocessing: we did not exclude stopwords and special characters (such as, for example, quotes), because they are quite often related to the propaganda class; we tried contexts of different sizes as the input, however, the context of 3 sentences turned out to be the most beneficial for our solution.

3.3 Implementation

We trained our models with Nvidia RTX 2080 Ti graphic cards. Our best solution was based on BERT-Base, Cased model. We used Adam optimizer with a learning rate of $3 \cdot 10^{-5}$. We fine-tuned such hyperparameters as the number of epochs, batch size, maximum length of sequence with Facebook Ax[4] library. For our best solution we chose the number of epochs 7, batch size 16, and sequence length 120. As we were only provided with the train set, we trained models with 3-Fold cross-validation.

3.4 Results

Submission results are presented in Table 1. Unfortunately, for BiLSTM-CNN-CRF model the amount of data was not enough. Although the solution based on this model overcame the baseline, it showed the worst result. BERT-CRF showed the best Precision but lost a few points in Recall. BERT achieved the best Recall as well as the F1 score outperforming the baseline by a large margin. The disadvantage of this model was that it did not combine neighboring words that could have been as a single phrase related to propaganda, which, in theory, BERT-CRF should have done. One of the approaches was to artificially combine nearby words into a single span. However, the best solution came without such post-processing.

4 Data Augmentation

We hypothesized that relatively low results obtained by the baseline model could be due to the reasons that, (i) one the one hand the semantics of the phenomenon at hand is complex and (ii) on the other hand, the training dataset is too small for even fine-tuning. Therefore we decided to perform automatic augmentation on the provided dataset to try reaching better generalization and more stable training of the baseline model.

4.1 Hypothesis

The hypothesis tested in our experiments was the following: the increase of the articles number with different data augmentation techniques will help to achieve a better generalization of the model due to more diverse training examples.

4.2 Method

We focused on the model that gave us the best F1 score on the dev set leaderboard. We tried several strategies for dataset expansions:

[4] https://github.com/facebook/Ax

Original	Even though the number of those infected has dropped in recent weeks, the plague will never truly be gone.
GloVe	
n	Even though the several of those infected has dropped in recent ago, the outbreak will never truly be gone.
n, adj	Even though the one of those infected has dropped in earlier last, the pneumonic will never truly be gone.
n, adj, adv	Even though the other of those infected has dropped in last month, the cholera will ever indeed be gone.
n, adj, adv, v	Even though the some of those hiv has slipped in earlier days, the bubonic will once quite be nothing.
fastText	
n	Even though the total of those infected has dropped in recent days, the pestilence will never truly be gone.
n, adj	Even though the amount of those infected has dropped in latest month, the scourge will never truly be gone.
n, adj, adv	Even though the size of those infected has dropped in last years, the bubonic will seldom hardly be gone.
n, adj, adv, v	Even though the quantity of those infested has fell in previous hours, the epidemic will rarely fully be went.
BERT	
n	Even though the majority of those infected has dropped in recent years, the disease will never truly be gone.
n, adj	Even though the fate of those infected has dropped in three decades, the infection will never truly be gone.
n, adj, adv	Even though the population of those infected has dropped in two months, the virus will now really be gone.
n, adj, adv, v	Even though the percentage of those dead has been in these times, the epidemic will soon fully be over.

Table 2: Example of our sentence augmentation method based on different (1) models for word embeddings, e.g. GloVe or BERT; (2) word POS used for that substitution, e.g. "n" for noun expansion. Red color denotes replaced nouns, green is adjectives, violet is adverbs, and blue color denotes replaced verbs. Finally, yellow box denotes the target propaganda span annotation.

- **Substitution model**. In order to find a replacement for the word, we used the search for the nearest word vector representations. In this research we decided to investigate research on *GloVe* (Pennington et al., 2014), *fastText* (Bojanowski et al., 2017) and *BERT* (Devlin et al., 2019) word embeddings.

- **Choice of words to replace**. We chose candidates for replacement based on their parts of speech (POS). At the same time, we did not replace stop words, as well as words with a high frequency of occurrence in the language. This was done not to replace the pronouns, common nouns (*everything, nothing*), numerals, common adverbs (*very*), etc. As a results, we combine several strategies for substitution: only nouns *(n)*; nouns and adjectives *(n, adj)*; nouns, adjectives, and adverbs *(n, adj, adv)*; nouns, adjectives, adverbs and verbs *(n, adj, adv, v)*.

- **Classes**. We also created combinations based on classes from which we chose words to substitute: only from propaganda class, only from neutral, or from both.

- **The increase of dataset ratio**. Another tested parameter is the output size of the final augmented dataset. We ran experiments with making two *(x2)*, five *(x5)* and ten *(x10)* fold augmentation. The increase of the dataset was done as follows: 1) the corresponding number of times the substitution algorithm was run for a sentence; 2) from the sentence at each run iteration 70% words from all candidates were randomly selected for substitution[5]; 3) for the selected words a replacement was randomly chosen from the top-5 list of the synonyms. These manipulations allowed us to obtain various combinations of substitutions, and, accordingly, more diverse contexts in the data.

[5]We selected the ratio of 70% to empirically based on the observation of the generated samples. The remaining 30% of words candidates on each iteration remained unchanged.

				F1	Precision	Recall			
			Baseline	0.3670	0.3498	0.3858			
	x2			x5			x10		
	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall
Substitution for:				**Propaganda class**					
GloVe									
n	0.3586	0.3477	0.3701	0.3454	0.3483	0.3425	0.3249	**0.3586**	0.2970
n, adj	0.3520	0.3385	0.3667	0.3543	**0.3741**	0.3364	0.3150	**0.3798**	0.2690
n, adj, adv	0.3567	0.3423	0.3724	0.3547	**0.3830**	0.3304	0.3361	**0.3621**	0.3135
n, adj, adv, v	0.3584	**0.3659**	0.3511	0.3439	**0.3629**	0.3269	0.3372	**0.3667**	0.3121
fastText									
n	0.3600	0.3392	0.3835	0.3430	**0.3703**	0.3195	0.3327	**0.3646**	0.3059
n, adj	0.3511	**0.3565**	0.3459	0.3446	**0.3671**	0.3247	0.3323	**0.3733**	0.2993
n, adj, adv	0.3433	0.3466	0.3401	0.3457	**0.3668**	0.3269	0.3311	**0.3631**	0.3042
n, adj, adv, v	0.3598	0.3383	0.3841	0.3280	**0.3556**	0.3044	0.3201	**0.3587**	0.2891
BERT									
n	0.3567	**0.3511**	0.3625	0.3288	0.3411	0.3174	0.3117	0.3331	0.2929
n, adj	0.3473	0.3412	0.3537	0.3407	**0.3545**	0.3280	0.3004	0.3420	0.2679
n, adj, adv	0.3429	0.3516	0.3347	0.3114	0.3448	0.2839	0.3043	0.3457	0.2717
n, adj, adv, v	0.3398	0.3252	0.3557	0.3268	0.3344	0.3196	0.3091	0.3396	0.2836
Substitution for:				**Neutral class**					
GloVe									
n	0.3590	0.3186	**0.4112**	0.3430	0.3196	0.3702	0.3082	0.3153	0.3014
n, adj	0.3594	0.3230	**0.4051**	0.3417	0.3261	0.3589	0.3273	0.3166	0.3388
n, adj, adv	0.3499	0.2980	**0.4237**	0.3275	0.3077	0.3501	0.3005	0.3020	0.2989
n, adj, adv, v	**0.3683**	0.3219	**0.4302**	0.3220	0.3117	0.3330	0.3300	0.3166	0.3445
fastText									
n	0.3500	0.3166	**0.3913**	0.3317	0.3148	0.3506	0.3309	0.3371	0.3249
n, adj	0.3652	0.3308	**0.4076**	0.3345	0.2992	0.3792	0.3145	0.3140	0.3150
n, adj, adv	0.3564	0.3305	**0.3867**	0.3354	0.3317	0.3392	0.3149	0.3026	0.3282
n, adj, adv, v	0.3560	0.3238	**0.3954**	0.3372	0.3340	0.3404	0.3194	0.3235	0.3154
BERT									
n	0.3254	0.3336	0.3176	0.3103	0.3467	0.2808	0.2825	0.3457	0.2389
n, adj	0.3450	0.3386	0.3517	0.3090	**0.3813**	0.2598	0.2574	**0.3726**	0.1966
n, adj, adv	0.3521	0.3494	0.3548	0.3161	**0.3714**	0.2751	0.3008	**0.3784**	0.2496
n, adj, adv, v	0.3505	0.3460	0.3550	0.3159	**0.3644**	0.2787	0.2642	**0.3507**	0.2120
Substitution for:				**Both classes**					
Glove									
n	0.3526	0.3335	0.3739	0.3432	**0.3552**	0.3321	0.3298	**0.3545**	0.3083
n, adj	0.3597	**0.3579**	0.3615	0.3377	**0.3587**	0.3190	0.3273	**0.3602**	0.2998
n, adj, adv	**0.3674**	**0.3677**	0.3670	0.3390	**0.3637**	0.3175	0.3228	0.3356	0.3110
n, adj, adv, v	0.3589	0.3577	0.3601	0.3458	0.3438	0.3478	0.3381	**0.3693**	0.3117
fastText									
n	0.3496	0.3451	0.3542	0.3489	**0.3674**	0.3322	0.3290	0.3495	0.3108
n, adj	0.3588	0.3373	0.3832	0.3413	**0.3600**	0.3244	0.3261	**0.3666**	0.2936
n, adj, adv	0.3608	**0.3610**	0.3605	0.3291	**0.3587**	0.3040	0.3165	**0.3764**	0.2731
n, adj, adv, v	0.3537	**0.3716**	0.3375	0.3371	**0.3637**	0.3141	0.3402	**0.3868**	0.3036
BERT									
n	0.3454	0.3363	0.3550	0.3236	**0.3552**	0.2971	0.3279	0.3380	0.3185
n, adj	0.3444	0.3499	0.3392	0.3231	0.3442	0.3044	0.3075	**0.3535**	0.2721
n, adj, adv	0.3455	0.3436	0.3475	0.3219	0.3466	0.3004	0.3077	0.3327	0.2862
n, adj, adv, v	0.3412	0.3186	0.3673	0.3086	0.3359	0.2854	0.3134	**0.3502**	0.2836

Table 3: Results of our augmentation strategies on the development set varying by the following parameters: (1) the number of times the dataset was increased, e.g. "x5" for five-fold expansion; (2) the class that took part in substitution – one can expand words either from "Propaganda" class, "Neural" class or both of them; (3) word vector model for synonyms search, e.g. GloVe; (4) POS used for that substitution, e.g. "n" for noun expansion or "adj" for adjectives. The top row shows result of the baseline BERT-based sequence tagger trained on the original dataset. The bold font denotes improvements with respect to this baseline while the underlined text denotes the best results outperformed the baseline.

4.3 Results

An example of our sentence augmentation method is presented in Table 2. We can see several quite successful replacements: for instance, the word *plague* was substituted with synonyms *cholera*, *epidemic*. Although substitutions are not always accurate in context, in general, the meaning of the sentence is preserved.

As BERT-Linear model showed the best results on dev set, we decided to focus on this model in our experiments. The results on dev set submissions for BERT-Linear model trained on augmented datasets are presented in Table 3. Expansion of neutral class allows us to boost Recall, and in some cases even without the loss of Precision (e.g. Glove, *x2*). Using this strategy we got the result better than the selected baseline ($F1 = 0.3683$). The increase of Recall is also observed when using fastText with replacing all parts of speech. So, it is disadvantageously to perform the expansions to nouns only – the majority of improvements occurred in POS combinations. In the case of Propaganda and Both classes, augmentation improves Precision of the model, especially when large number (*x5*, *x10*) of expansions is performed. BERT-based expansions show worse results than Glove and fastText. The reason for that can be the availability of sufficient information about vector embeddings in the language model itself.

Therefore, the following conclusion can be made: the increase of dataset with several augmentation strategies, unfortunately, did not give a strong improvement to the model performance. However, some applied methods for data extension gave a significant improvement in Recall metric.

5 Conclusion

We presented the solution of "SkoltechNLP" team for the Span Identification task in the SemEval-2020 task 11 competition. Our final solution is based on the BERT masked language model, specially pretrained for the NER task, which showed strong performance out-of-the-box with respect to the baseline. In addition, we investigated various strategies for the dataset augmentation on the public set of our best model, trained on the expanded text datasets. Unfortunately, this approach did not give a significant increase of the F1 score. However, it was shown that the proposed strategies can substantially improve precision if words from the target "Propaganda" class are expanded and improve recall if substitutions for the neutral class is used to generate new training examples. Therefore, the developed expansion methods could be useful for shifting "sweet spot" of a classification model between precison and recall maintaining similar F1 level.

As future work, more careful search on hyperparameters for augmentation can be considered. For example, in this work we just manually selected the ratio of dataset increase (*x2*, *x5*, *x10*), however, this parameter can be searched on the scale of natural numbers. The same can be done with the ratio of words selected for substitution. We selected 70% ratio from our own conclusions of the diversity of the new contexts obtained, but this number can also vary. Another aspect not covered in the work is the imbalance of classes. In this case, expanding the data to balance examples in both classes, as well as redistributing class weights, can be useful for obtaining a more stable model.

Acknowledgements

We thank Nikolay Arefyev and Artem Shelmanov for helpful suggestions on the present study.

References

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357.

Artem Chernodub, Oleksiy Oliynyk, Philipp Heidenreich, Alexander Bondarenko, Matthias Hagen, Chris Biemann, and Alexander Panchenko. 2019. TARGER: Neural argument mining at your fingertips. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 195–200, Florence, Italy, July. Association for Computational Linguistics.

Giovanni Da San Martino, Alberto Barron-Cedeno, and Preslav Nakov. 2019. Findings of the nlp4if-2019 shared task on fine-grained propaganda detection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 162–170.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June. Association for Computational Linguistics.

Adam Ek and Mehdi Ghanimifard. 2019. Synthetic propaganda embeddings to train a linear projection. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 155–161.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany, August. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June. Association for Computational Linguistics.

Artem Shelmanov, Vadim Liventsev, Danil Kireev, Nikita Khromov, Alexander Panchenko, Irina Fedulova, and Dmitry V. Dylov. 2019. Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts. In *Proceedings of International Conference on Bioinformatics Biomedicine (BIBM)*.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Shehel Yoosuf and Yin Yang. 2019. Fine-grained propaganda detection with fine-tuned bert. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, pages 87–91.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. Defending against neural fake news. In *Advances in Neural Information Processing Systems*, pages 9051–9062.