InfoMiner at WNUT-2020 Task 2: Transformer-based Covid-19 Informative Tweet Extraction

Hansi Hettiarachchi[♡], **Tharindu Ranasinghe**[‡]
[♡]School of Computing and Digital Technology, Birmingham City University, UK
[‡]Research Group in Computational Linguistics, University of Wolverhampton, UK
hansi.hettiarachchi@mail.bcu.ac.uk
tharindu.ranasinghe@wlv.ac.uk

Abstract

Identifying informative tweets is an important step when building information extraction systems based on social media. WNUT-2020 Task 2 was organised to recognise informative tweets from noise tweets. In this paper, we present our approach to tackle the task objective using transformers. Overall, our approach achieves 10^{th} place in the final rankings scoring 0.9004 F1 score for the test set.

1 Introduction

By 31st August 2020, coronavirus COVID-19 is affecting 213 countries around the world infecting more than 25 million people and killing more than 800,000. Recently, much attention has been given to build monitoring systems to track the outbreaks of the virus. However, due to the fact that most of the official news sources update the outbreak information only once or twice a day, these monitoring tools have begun to use social media as the medium to get information.

There is a massive amount of data on social networks, e.g. about 4 millions of COVID-19 English tweets daily on the Twitter platform. However, majority of these tweets are uninformative. Thus it is important to be able to select the informative ones for downstream applications. Since the manual approaches to identify the informative tweets require significant human efforts, an automated technique to identify the informative tweets will be invaluable to the community.

The objective of this shared task is to automatically identify whether a COVID-19 English tweet is informative or not. Such informative Tweets provide information about recovered, suspected, confirmed and death cases as well as location or travel history of the cases. The participants of the shared task were required to provide predictions for the test set provided by the organisers whether a tweet is informative or not. Our team used recently released transformers to tackle the problem. Despite achieving 10^{th} place out of 55 participants and getting high evaluation score, our approach is simple and efficient. In this paper we mainly present our approach that we used in this task. We also provide important resources to the community: the code, and the trained classification models will be freely available to everyone interested in working on identifying informative tweets using the same methodology [1].

2 Related Work

In the last few years, there have been several studies published on the application of computational methods in order to identify informative contents from tweets. Most of the earlier methods were based on traditional machine learning models like logistic regression and support vector machines with heavy feature engineering. Castillo et al. (2011) investigate tweet newsworthiness classification using features representing the message, user, topic and the propagation of messages. Others use features based on social influence, information propagation, syntactic and combinations of local linguistic features as well as user history and user opinion to select informative tweets (Inouye and Kalita, 2011; Yang et al., 2011; Chua and Asur, 2013). Due to the fact that training set preparation is difficult when it comes informative tweet identification, several studies suggested unsupervised methods. Sankaranarayanan et al. (2009) built a news processing system, called *TwitterStand* using an unsupervised approach to classify tweets collected from pre-determined users who frequently post news about events. Even though these traditional approaches have provided good results, they

[1]The GitHub repository is publicly available on https://github.com/hhansi/informative-tweet-identification

Proceedings of the 2020 EMNLP Workshop W-NUT: The Sixth Workshop on Noisy User-generated Text, pages 359–365

are no longer the state of the art.

Considering the recent research, there was a tendency to use deep learning-based methods to identify informative tweets since they performed better than traditional machine learning-based methods. To mention few, ALRashdi and O'Keefe (2019) suggested an approach based on Bidirectional Long Short-Term Memory (Bi-LSTM) models trained using word embeddings. Another research proposed a deep multi-modal neural network based on images and text in tweets to recognise informative tweets (Kumar et al., 2020). Among the different neural network models available, transformer models received a huge success in the area of natural language processing (NLP) recently. Since the release of BERT (Devlin et al., 2019), transformer models gained a wide attention of the community and they were successfully applied for wide range of tasks including tweet classification tasks such as offensive tweet identification (Ranasinghe et al., 2019) and topic identification (Yüksel et al., 2019). But we could not find any previous work on transformers for informative tweet classification. Hence, we decided to use transformer for our approach and this study will be important to the community.

3 Task Description and Data Set

WNUT-2020 Task 2: Identification of informative COVID-19 English Tweets (Nguyen et al., 2020) is to develop a system which can automatically categorise the tweets related to coronavirus as informative or not. A data set of 10K tweets which are labelled as *informative* and *uninformative* is released to conduct this task. The class distributions of the data set splits are mentioned in Table 1.

Data set	Informative	Uninformative
Training	3303	3697
Validation	472	528
Test	944	1056

Table 1: Class distribution of data set splits

4 Methodology

The motivation behind our methodology is the recent success that the transformers had in wide range of NLP tasks like language generation (Devlin et al., 2019), sequence classification (Ranasinghe and Hettiarachchi, 2020; Ranasinghe et al., 2019; Ranasinghe and Zampieri, 2020), word similarity (Hettiarachchi and Ranasinghe, 2020), named entity recognition (Liang et al., 2020) and question and answering (Yang et al., 2019a). The main idea of the methodology is that we train a classification model with several transformer models in-order to identify informative tweets.

4.1 Transformers for Text Classification

Predicting whether a certain tweet is informative or not can be considered as a sequence classification task. Since the transformer architectures have shown promising results in sequence classification tasks (Ranasinghe and Hettiarachchi, 2020; Ranasinghe et al., 2019; Ranasinghe and Zampieri, 2020), the basis for our methodology was transformers. Transformer architectures have been trained on general tasks like language modelling and then can be fine-tuned for classification tasks. (Sun et al., 2019)

Transformer models take an input of a sequence and outputs the representations of the sequence. There can be one or two segments in a sequence which are separated by a special token [SEP]. In this approach we considered a tweet as a sequence and no [SEP] token is used. Another special token [CLS] is used as the first token of the sequence which contains a special classification embedding. For text classification tasks, transformer models take the final hidden state $\mathbf{h}$ of the [CLS] token as the representation of the whole sequence (Sun et al., 2019). A simple softmax classifier is added to the top of the transformer model to predict the probability of a class c as shown in Equation 1 where W is the task-specific parameter matrix. The architecture of transformer-based sequence classifier is shown in Figure 1.

$$p(c|\mathbf{h}) = softmax(W\mathbf{h}) \quad (1)$$

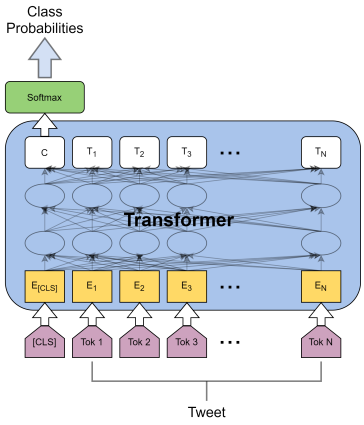

Figure 1: Text Classification Architecture

4.2 Transformers

We used several pre-trained transformer models in this task. These models were used mainly considering the popularity of them (e.g. BERT (Devlin et al., 2019), XLNet (Yang et al., 2019b), RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020), ALBERT (Lan et al., 2020)) and relatedness to the task (e.g. COVID-Twitter-BERT (CT-BERT) (Müller et al., 2020) and BERTweet (Dat Quoc Nguyen and Nguyen, 2020)).

BERT (Devlin et al., 2019) was the first transformer model that gained a wide attention of the NLP community. It proposes a masked language modelling (MLM) objective, where some of the tokens of a input sequence are randomly masked, and the objective is to predict these masked positions taking the corrupted sequence as input. As we explained before BERT uses special tokens to obtain a single contiguous sequence for each input sequence. Specifically, the first token is always a special classification token [CLS] which is used for sentence-level tasks.

RoBERTa (Liu et al., 2019), ELECTRA (Clark et al., 2020) and ALBERT (Lan et al., 2020) can all be considered as variants of BERT. They make a few changes to the BERT model and achieves substantial improvements in some NLP tasks (Liu et al., 2019; Clark et al., 2020; Lan et al., 2020). XLNet on the other hand takes a different approach to BERT (Yang et al., 2019b). XLNet proposes a new auto-regressive method based on permutation language modelling (PLM) (Uria et al., 2016) without introducing any new symbols such as [MASK] in BERT. Also there are significant changes in the XLNet architecture like adopting two-stream self-attention and Transformer-XL (Dai et al., 2019). Due to this XLNet outperforms BERT in multiple NLP downstream tasks (Yang et al., 2019b).

We also used two transformer models based on Twitter; CT-BERT and BERTweet. The CT-BERT model is based on the BERT-LARGE model and trained on a corpus of 160M tweets about the coronavirus (Müller et al., 2020) while the BERTweet model is based on BERT-BASE model and trained on general tweets (Dat Quoc Nguyen and Nguyen, 2020).

4.3 Data Preprocessing

Few general data preprocessing techniques were employed with InfoMiner to preserve the universality of this method. More specifically, used techniques can be listed as removing or filling usernames and URLs, and converting emojis to text. Further, for uncased pretrained models (e.g. *albert-xxlarge-v1*), all tokens were converted to lower case.

In WNUT-2020 Task 2 data set, mention of a user is represented by *@USER* and a URL is represented by *HTTPURL*. For all the models except CT-BERT and BERTweet, we removed those mentions. The main reason behind this step is to remove noisy text from data. CT-BERT and BERTweet models are trained on tweet corpora and usernames and URLs are introduced to the models using special fillers. CT-BERT model knows a username as *twitteruser* and URL as *twitterurl*. Likewise, BERTweet model used the filler *@USER* for usernames and *HTTPURL* for URLs. Therefore, for these two models we used the corresponding fillers to replace usernames and URLs in the data set.

Emojis are found to play a key role in expressing emotions in the context of social media (Hettiarachchi and Ranasinghe, 2019). But, we cannot assure the existence of embeddings for emojis in pretrained models. Therefore as another essential preprocessing step, we converted emojis to text. For this conversion we used the Python libraries *demoji*[2] and *emoji*[3]. *demoji* returns a normal descriptive text and *emoji* returns a specifically formatted text. For an example, the conversion of ☺ is 'slightly smiling face' using *demoji* and ':slightly_smiling_face:' using *emoji*. For all the models except CT-BERT and BERTweet, we used *demoji* supported conversion. For CT-BERT and BERTweet *emoji* supported conversion is used, because these models are trained on correspondingly converted Tweets.

4.4 Fine-tuning

To improve the models, we experimented different fine-tuning strategies: majority class self-ensemble, average self-ensemble, entity integration and language modelling, which are described below.

1. **Self-Ensemble (SE)** - Self-ensemble is found as a technique which result better performance than the performance of a single model (Xu et al., 2020). In this approach, same model architecture is trained or fine-tuned with different random seeds or train-validation splits.

[2]demoji repository - https://github.com/bsolomon1124/demojis

[3]emoji repository - https://github.com/carpedm20/emoji

Strategy	Single-model			MSE (N=3)		
Model	**P**	**R**	**F1**	**P**	**R**	**F1**
bert-large-cased	0.9031	0.8686	0.8855	0.8884	0.8941	0.8912
roberta-large	0.9056	0.8941	0.8998	0.8926	0.9153	0.9038
albert-xxlarge-v1	0.9009	0.8856	0.8932	0.9032	0.8898	0.8965
xlnet-large-cased	0.8778	0.9280	0.9022	0.8743	0.9280	0.9003
electra-large-generator	0.8297	0.8771	0.8527	0.8901	0.8750	0.8825
bertweet-base	0.8710	0.8968	0.8753	0.8741	0.8998	0.8780
covid-twitter-bert	0.8984	0.9364	0.9170	0.9002	0.9364	0.9180

Table 2: Results of different transformer models (All these experiments are executed for 3 learning epochs with $1e^{-5}$ learning rate.)

Learning R.		$1e^{-5}$			$1e^{-6}$			$2e^{-5}$		
S. 1	**S. 2**	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
MSE (N=3)	-	0.9072	0.9322	0.9195	0.9317	0.8962	0.9136	0.9125	0.9280	0.9202
	EI	0.8864	0.9258	0.9057	0.9181	0.9025	0.9103	0.8975	0.9089	0.9032
	LM	0.8912	0.9195	0.9051	0.8987	0.9025	0.9006	0.9070	0.9301	0.9184
ASE (N=3)	-	0.9091	0.9322	0.9205	0.9295	0.8941	0.9114	0.9146	0.9301	**0.9223**
	EI	0.8960	0.9131	0.9045	0.9124	0.9047	0.9085	0.9025	0.9025	0.9025
	LM	0.9021	0.9174	0.9097	0.8971	0.9047	0.9008	0.9160	0.9237	0.9198

Table 3: Result obtained for CT-BERT model with different fine-tuning strategies (All these experiments are executed for 5 learning epochs and S. abbreviates the Strategy)

Then the output of each model is aggregated to generate the final results. As the aggregation methods, we analysed majority-class and average in this research. The number of models used with self-ensemble will be denoted by N.

- *Majority-class SE (MSE)* - As the majority class, we computed the mode of the classes predicted by each model. Given a data instance, following the softmax layer, a model predicts probabilities for each class and the class with highest probability is taken as the model predicted class.
- *Average SE (ASE)* - In average SE, final probability of class c is calculated as the average of probabilities predicted by each model as in Equation 2 where *h* is the final hidden state of the [CLS] token. Then the class with highest probability is selected as the final class.

$$p_{ASE}(c|h) = \frac{\sum_{k=1}^{N} p_k(c|h)}{N} \quad (2)$$

2. **Entity Integration (EI)** - Since we are using pretrained models, there can be model unknown data in the task data set such as person names, locations and organisations. As entity integration, we replaced the unknown tokens with their named entities which are known to the model, so that the familiarity of data to model can be increased. To identify the named entities, we used the pretrained models available with spaCy [4].

3. **Language Modelling (LM)** - As language modelling, we retrained the transformer model on task data set before fine-tuning it for the downstream task; text classification. This training is took place according with the model's initial trained objective. Following this technique model understanding on the task data can be improved.

5 Experiments and Results

In this section, we report the experiments we conducted and their results. As informed by task organisers, we used precision, recall and F1 score calculated for *Informative* class to measure the model performance. Results in sections 5.1 - 5.3 are computed on validation data set and results in section

[4]More details about spaCy are available on https://spacy.io/

5.4 are computed on test data set.

5.1 Impact by Transformer Model

Initially we focused on the impact by different transformer models. Selected transformer models were fine-tuned for this task using single-model (no ensemble) and MSE with 3 models, and the obtained results are summarised in Table 2. According to the results, CT-BERT model outperformed the other models. Also, all the models except XL-Net showed improved results with self-ensemble approach than single-model approach. Following these results and considering time and resource constraints, we limited the further experiments only to CT-BERT model.

5.2 Impact by Epoch Count

We experimented that increasing the epoch count from 3 to 5 increases the results. However, increasing it more than 5 did not further improved the results. Therefore, we used an epoch count of 5 in our experiments. To monitor the evaluation scores against the epoch count we used Wandb app [5]. As shown in the Figure 2 evaluation f1 score does not likely to change when trained with more than five epochs.

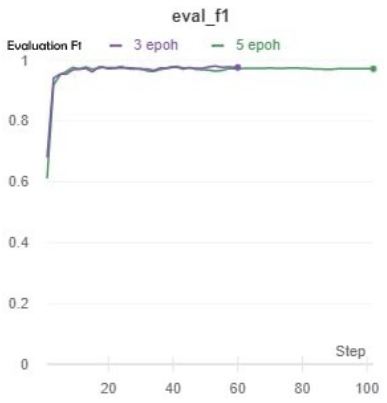

Figure 2: Evaluation F1 score against the epoch count

5.3 Impact by Fine-tuning

The fine-tuning strategies mentioned in Section 4.4 were experimented using CT-BERT model and obtained results are summarised in Table 3. According to the results, in majority of experiments, ASE is given a higher F1 than MSE. The other fine-strategies: EI and LM did not improve the results for this data set. As possible reasons for this reduction, having a good knowledge about COVID tweets by the model itself and insufficiency of data for language modelling can be mentioned.

Additionally, we analysed the impact by different learning rates. For initial experiments a random learning rate of $1e^{-5}$ was picked and for further analysis a less value ($1e^{-6}$) and a high value ($2e^{-5}$) were picked. The value $2e^{-5}$ was used for pretraining and experiments of CT-BERT model (Müller et al., 2020). According to this analysis there is a tendency to have higher F1 with higher learning rates.

5.4 Test Set Evaluation

The test data results of our submissions, task baseline and top-ranked system are summarised in Table 4. Considering the evaluation results on validation data set, as InfoMiner 1 we selected the fine-tuned CT-BERT model with ASE and $2e^{-5}$ learning rate. As InfoMiner 2 same model and parameters with MSE was picked. Among them, the highest F1 we received is for MSE strategy.

Model	P	R	F1
Top-ranked	0.9135	0.9057	0.9096
InfoMiner 1	0.9107	0.8856	0.8980
InfoMiner 2	0.9102	0.8909	0.9004
Task baseline	0.7730	0.7288	0.7503

Table 4: Results of test data predictions

6 Conclusion

We have presented the system by InfoMiner team for WNUT-2020 Task 2. For this task, we have shown that the CT-BERT is the most successful transformer model from several transformer models we experimented. Furthermore, we presented several fine tuning strategies: self-ensemble, entity integration and language modelling that can improve the results. Overall, our approach is simple but can be considered as effective since it achieved 10^{th} place in the leader-board.

As a future direction of this research, we hope to analyse the impact by different classification heads such as LSTM and Convolution Neural Network (CNN) in addition to softmax classifier on performance. Also, we hope to incorporate meta information-based features like number of retweets and likes with currently used textual features to involve social aspect for informative tweet identification.

[5]Wandb app is available on https://app.wandb.ai/

References

Reem ALRashdi and Simon O'Keefe. 2019. Deep learning and word embeddings for tweet classification for crisis response. *arXiv preprint arXiv:1903.11024*.

Carlos Castillo, Marcelo Mendoza, and Barbara Poblete. 2011. Information credibility on twitter. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, page 675–684, New York, NY, USA. Association for Computing Machinery.

Freddy Chua and Sitaram Asur. 2013. Automatic summarization of events from social media.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *ICLR*.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Thanh Vu Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. *arXiv preprint*, arXiv:2005.10200.

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. Association for Computational Linguistics.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2019. Emoji powered capsule network to detect type and target of offensive posts in social media. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 474–480.

Hansi Hettiarachchi and Tharindu Ranasinghe. 2020. Brums at semeval-2020 task 3: Contextualised embeddings for predicting the (graded) effect of context in word similarity. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

D. Inouye and J. K. Kalita. 2011. Comparing twitter summarization algorithms for multiple post summaries. In *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, pages 298–306.

Abhinav Kumar, Jyoti Prakash Singh, Yogesh K. Dwivedi, and Nripendra P. Rana. 2020. A deep multi-modal neural network for informative twitter content classification during emergencies. *Annals of Operations Research*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '20, page 1054–1064, New York, NY, USA. Association for Computing Machinery.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Martin Müller, Marcel Salathé, and Per E Kummervold. 2020. Covid-twitter-bert: A natural language processing model to analyse covid-19 content on twitter. *arXiv preprint arXiv:2005.07503*.

Dat Quoc Nguyen, Thanh Vu, Afshin Rahimi, Mai Hoang Dao, Linh The Nguyen, and Long Doan. 2020. WNUT-2020 Task 2: Identification of Informative COVID-19 English Tweets. In *Proceedings of the 6th Workshop on Noisy User-generated Text*.

Tharindu Ranasinghe and Hansi Hettiarachchi. 2020. BRUMS at SemEval-2020 task 12 : Transformer based multilingual offensive language identification in social media. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Tharindu Ranasinghe and Marcos Zampieri. 2020. Multilingual offensive language identification with cross-lingual embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Tharindu Ranasinghe, Marcos Zampieri, and Hansi Hettiarachchi. 2019. BRUMS at HASOC 2019: Deep learning models for multilingual hate speech and offensive language identification. *In Proceedings of the 11th annual meeting of the Forum for Information Retrieval Evaluation (December 2019)*.

Jagan Sankaranarayanan, Hanan Samet, Benjamin E. Teitler, Michael D. Lieberman, and Jon Sperling. 2009. Twitterstand: News in tweets. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS '09, page 42–51, New York, NY, USA. Association for Computing Machinery.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *Chinese Computational Linguistics*, pages 194–206, Cham. Springer International Publishing.

Benigno Uria, Marc-Alexandre Côté, Karol Gregor, Iain Murray, and Hugo Larochelle. 2016. Neural autoregressive distribution estimation. *J. Mach. Learn. Res.*, 17(1):7184–7220.

Yige Xu, Xipeng Qiu, Ligao Zhou, and Xuanjing Huang. 2020. Improving bert fine-tuning via self-ensemble and self-distillation. *arXiv preprint arXiv:2002.10345*.

Wei Yang, Yuqing Xie, Aileen Lin, Xingyu Li, Luchen Tan, Kun Xiong, Ming Li, and Jimmy Lin. 2019a. End-to-end open-domain question answering with BERTserini. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 72–77, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zi Yang, Keke Cai, Jie Tang, Li Zhang, Zhong Su, and Juanzi Li. 2011. Social context summarization. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '11, page 255–264, New York, NY, USA. Association for Computing Machinery.

Atıf Emre Yüksel, Yaşar Alim Türkmen, Arzucan Özgür, and Berna Altınel. 2019. Turkish tweet classification with transformer encoder. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1380–1387.