

# ComplexDataLab at WNUT-2020 Task 2: Detecting Informative COVID-19 Tweets by Attending over Linked Documents

Kellin Pelrine, Jacob Danovitch, Albert Orozco Camacho, Reihaneh Rabbany

School of Computer Science, McGill University

Mila - Quebec AI Institute

{kellin.pelrine, jacob.danovitch, alorozco53, reihaneh.rabbany}@mila.quebec

## Abstract

Given the global scale of COVID-19 and the flood of social media content related to it, how can we find informative discussions? We present GAPFORMER, which effectively classifies content as informative or not. It reformulates the problem as graph classification, drawing on not only the tweet but connected web pages and entities. We leverage a pre-trained language model as well as the connections between nodes to learn a pooled representation for each document network. We show it outperforms several competitive baselines and present ablation studies supporting the benefit of the linked information. Code is available on [GitHub](#)<sup>1</sup>.

## 1 Introduction

COVID-19 is a critical public health crisis, with over 25 million cases and 840 thousand deaths worldwide and counting (WHO, 2020). One important way to fight the pandemic is through understanding and leveraging the vast amount of information humans produce related to it on social media. Success here can facilitate individual and overall case detections, contact tracing, case predictions, effective information dissemination, and more (Li et al., 2020; Qin et al., 2020). However, the data is far too big - for example, hundreds of millions of tweets (Qazi et al., 2020) - to extract all the useful information by hand. There is also a huge amount of useless and even outright false information (Alam et al., 2020; Cui and Lee, 2020). Therefore, algorithms that extract useful information effectively are crucial.

WNUT-2020 Task 2 (Nguyen et al., 2020) is to build such a system on a particular twitter dataset, and predict which tweets are informative “about

<sup>1</sup><https://github.com/ComplexData-MILA/gapformer>

recovered, suspected, confirmed and death cases as well as location or travel history of the cases.” We present here GAPFORMER, which combines multiple sources of information - the tweet, content of linked web pages, and information about named entities related to the tweet - to improve detection of informative content.

Our experiments show GAPFORMER outperforms 7 baselines. It also performs best when incorporating all three types of data, showing the efficacy of using the proposed graph structure. We believe this method can also be applied in other related settings, such as misinformation detection, and have work in progress to investigate possible extensions.

## 2 Background and Related Work

Before COVID-19, mining information from tweets about disaster events has been considered in several contexts. The CrisisNLP (Imran et al., 2016) collection has twitter datasets from 19 crises, including diseases Ebola and MERS. There are also works analyzing particular events (Takahashi et al., 2015; Kankanamge et al., 2020; Chen et al., 2020) and methods (Kumar et al., 2011; To et al., 2017; Singh et al., 2019; Aggarwal, 2019).

However, COVID-19 is unprecedented in its global scale and impact. While there is also a historic amount of research taking place to counter it (Brainard, 2020), more is needed - there are still lives to be saved. In addition, while people have tackled related COVID-19 twitter data tasks (Boulos and Geraghty, 2020; Alam et al., 2020; Cui and Lee, 2020), to our knowledge this particular task formulation has not been thoroughly investigated prior to this shared task competition.

With the success of transformer-based models (Vaswani et al., 2017; Devlin et al., 2018), it is natural to consider applying them here. BERTweet

(Dat Quoc Nguyen and Nguyen, 2020) and covid-twitter-bert (Müller et al., 2020), BERT language models fine-tuned on twitter data (and the latter on tweets related to COVID-19 specifically), are particularly relevant. However, pure language models struggle to consider additional contextual information beyond the raw content of the tweet. As our experiments show, this is an important limitation.

Because this limitation applies to many other applications as well, a rapidly growing area of research is embedding additional contextual knowledge into language models, especially with the use of graph structures (Rosset et al., 2020; Zhang, 2020; Cui et al., 2020). This is often done at the language modelling level, such as how (Lu et al., 2020) uses normalized mutual information to inject global information into each layer of BERT. That said, other work has focused on injecting knowledge into an existing pre-trained language model. In particular, Transformer-XH (Zhao et al., 2020) proposes a way to leverage the network structure of documents alongside a pre-trained model for a classification task. GAPFORMER builds on this line of work by proposing a simple yet effective architecture which contextualizes each node before pooling the graph into a single fixed embedding.

### 3 Proposed Method

Symbols	Definitions
$\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$	A graph and its vertex/edge sets
$\mathbf{A}$	Adjacency matrix of $\mathcal{G}$
$\mathbf{D}$	Degree vector of $\mathcal{V}$
LM	Pre-trained language model
$\mathbf{W}, \mathbf{b}$	Weight matrix, bias vector
$t$	A tweet
$A_t = \{a_i\}$	Articles linked to by $t$
$E_t = \{e_j\}$	Entities mentioned by $t$

Table 1: Symbols and Definitions

Our work presents **GrAph Pooling Transformer** (GAPFORMER), which combines powerful semantic representations from pretrained language models with structured information from graph neural networks by incorporating additional context from entities and web documents within each tweet. We reformulate the task as graph classification, where each instance is a single graph comprised of one tweet as well as the extracted entities and docu-

ments.

### 3.1 Graph Construction

For each instance, we construct a graph  $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$  with the documents and entities in each tweet  $t$  as nodes and  $t$  itself as a supernode. We describe the construction process below.

#### 3.1.1 Node Selection

We extract each entity mentioned in a given tweet  $t$ , and link them to their respective Wikipedia entries. We then represent the entities with their associated Wikipedia documents. The full set of entities extracted from  $t$  forms  $E_t$ . We use the Python packages FLAIR (Akbik et al., 2019) for named entity recognition and BLINK (Wu et al., 2019) for entity linking.

We also retrieve the content of any news articles linked to by the tweet, with the intuition that tweets which cite their sources are more likely to be informative, and that the content of said sources is likely to provide a useful signal as well. We use the newspaper3k Python package (Lucas Ou-Yang, 2013) to retrieve the summaries for each article, as full articles tend to be particularly long and may contain superfluous information. This set of articles forms  $A_t$ .

Finally, to form the vertex set, we take  $\mathcal{V} = A_t \cup E_t \cup \{t\}$ .

#### 3.1.2 Edge Selection

To begin, we draw an edge between  $t$  and each node  $n \in \mathcal{V}$  to embed  $t$  with additional information (including  $t$  itself, forming a self-loop). To better contextualize this information, we also draw edges for pairs of entities, as the intent of an entity mention may vary between different contexts. However, we must also be wary of simply adding as many connections as possible, which would decrease efficiency. We select which entity pairs to connect by calculating the normalized pointwise mutual information (NPMI) (Lu et al., 2020) between each pair. Using NPMI as the selection criterion allows us to connect nodes which frequently co-occur (mentioned in the same tweet) throughout the dataset, suggesting a more meaningful relationship. Empirically, we observed 0.15 to be a sensible threshold.

For entities  $e_i$  and  $e_j$ :

$$\text{pmi}(e_i, e_j) = \frac{p(e_i, e_j)}{p(e_i)p(e_j)} \quad (1)$$

$$\text{npmi}(e_i, e_j) = \frac{\text{pmi}(e_i, e_j)}{-\log p(x, y)} \quad (2)$$

The full algorithm for edge selection is described in algorithm 1.

```

Data: Vertex set  $\mathcal{V} = A_t \cup E_t \cup \{t\}$ 
Result: Edge set  $\mathcal{E}$ 
1  $\mathcal{E} = \{\}$ ;
2 for  $n$  in  $\mathcal{V}$  do
3   | add  $(n, t)$  to  $\mathcal{E}$ 
4 end
5 for  $e_i$  in  $E_t$  do
6   | for  $e_j$  in  $E_t$  do
7     | if  $\text{npmi}(e_i, e_j) > 0.15$  then
8       | | add  $(e_i, e_j)$  to  $\mathcal{E}$ 
9       | end
10  | end
11 end

```

**Algorithm 1:** Edge selection algorithm

### 3.2 GAPFORMER

For each instance in the dataset, we construct a graph  $\mathcal{G}$  as described above. Each node  $n_i \in \mathcal{G}$  is represented by a text document (either a tweet, entity description, or article). We tokenize each node and obtain word embeddings using a pre-trained transformer language model. We pool the word embeddings by selecting the embedding for the [CLS] tokens from the final output layer, forming the graph’s node embeddings.

After embedding each node with the pre-trained language model, we apply  $k$  layers of mean-pooling GraphSAGE convolutions (Hamilton et al., 2017) to contextualize each node with respect to their neighbors. We then aggregate the graph to a single embedding using max pooling, attention, or another pooling mechanism. Empirically, we find max pooling to be most effective on this dataset. Finally, we use a linear layer to predict output logits.

The full algorithm is presented in algorithm 2. The model is trained end-to-end using cross entropy loss, and optimized with AdamW (Loshchilov and Hutter, 2017). Full implementation details are available in our GitHub repository.

**Data:** Vertices  $\mathcal{V}$ ; Adjacency matrix  $\mathbf{A}$ ;  
Degree vector  $\mathbf{D}$

**Result:** Prediction logits  $\hat{\mathbf{y}}$

```

1  $\mathbf{x} = \text{LM}(\mathcal{V})$ ;
2 for  $i$  in  $1..k$  do
3   |  $\mathbf{h} = \mathbf{D}^{-1} \cdot \mathbf{A}\mathbf{x}$  ;
4   |  $\mathbf{x} = \sigma(\mathbf{W}_i \cdot \text{concat}(\mathbf{x}, \mathbf{h}) + \mathbf{b}_i)$  ;
5 end
6  $\mathbf{x} = \text{pooling}(\mathbf{x})$  ;
7  $\hat{\mathbf{y}} = \sigma(\mathbf{W}_\ell \mathbf{x} + \mathbf{b})$ 

```

**Algorithm 2:** GAPFORMER algorithm

## 4 Experiments

### 4.1 Data

The dataset provided for this task (Nguyen et al., 2020) consists of ten thousand tweets, labeled “informative” and “uninformative.” A default split into train, validate, and test sets is also given, with details shown in table 2.

	Informative	Uninformative	Total
<i>Train</i>	3303	3697	7000
<i>Validate</i>	472	528	1000
<i>Test</i>	944	1056	2000

Table 2: Dataset Information

Since we do not have access to the actual test set for this task (it will only be published after camera-ready deadline for this paper), we treat the validation set provided as the test set in our reported results. We also randomly split 1000 tweets from the train set as the validation set used for hyper-parameter tuning of the deep learning based models, i.e. early stopping, to avoid corrupting the actual validation set (our test set).

### 4.2 Baselines

We compare our model with several baselines:

- Naive Bayes (**NB**).
- Logistic Regression (**LR**).
- Support Vector Machine (**SVM**).
- GloVe + **LSTM**. This applies an LSTM to word embeddings generated from the tweets using 200-dimensional GloVe Twitter word embeddings (Pennington et al., 2014).
- Base **BERT**. This is the original BERT model with a classification head. The entire model is

fine-tuned during training. It is implemented using Huggingface Transformers and trained using Pytorch-Lightning. (Wolf et al., 2019; Falcon, 2019).

- **BERTweet** (Dat Quoc Nguyen and Nguyen, 2020). This is a BERT model fine-tuned on twitter data. Our implementation fine-tuned the weights provided by the authors with training identical to Base BERT.
- **COVID-Twitter-BERT (CTBERT)** (Müller et al., 2020). This is BERT but fine-tuned on COVID-19 twitter data. It was also trained identically to Base BERT.

The first 3 all use term frequency-inverse document frequency (TF-IDF) as features and the default Scikit-learn (Pedregosa et al., 2011) implementation. The last 3 (BERT-based models), as well as GAPFORMER, all use the AdamW optimizer (Loshchilov and Hutter, 2017) with a linear scheduler with warmup, set to learning rate  $7e-6$ , and are trained using an NVIDIA RTX8000 GPU. For GAPFORMER, we find **CTBERT** to be the most effective pre-trained language model, and use it in all experiments. We train it for 8 epochs with batch size 2 and accumulating 8 batches for gradient calculations, because CTBert is too big to fit a bigger batch in GPU memory. We also use precision 16, max sequence length 128, .5 dropout for the graph stage, and max pooling.

### 4.3 Results

	F1	Precision	Recall	Accuracy
<b>NB</b>	.743	.770	.718	.766
<b>LR</b>	.791	.840	.748	.814
<b>SVM</b>	.803	.858	.754	.825
<b>LSTM</b>	.823	.830	.816	.834
<b>BERT</b>	.894	.879	.909	.898
<b>BERTweet</b>	.903	.862	<b>.949</b>	.904
<b>CTBERT</b>	.913	.903	.924	.917
<b>GAPFORMER</b>	<b>.927</b>	<b>.915</b>	.939	<b>.930</b>

Table 3: Results

As shown in table 3, GAPFORMER outperforms the baselines in all categories except recall. The strength in precision matches with our results in the shared task competition (on the test set, rather than the withheld validation set used above), where we were #1 in that metric, as shown in table 4.

Metric	Value	Rank
Accuracy	.903	10
Precision	.920	1
Recall	.871	23
F-1	.895	17

Table 4: Shared task competition results. 122 teams total, of which 55 submitted results.

### 4.4 Effect of Articles and Entities

To examine the effectiveness of each component of our graph, we conduct an ablation study. We compare the full GAPFORMER using tweets, entities, and articles linked in the tweet (including other tweets if a URL points to them), denoted here **T + E + A**, with tweets only (**T**), tweets plus entities only (**T + E**), and tweets plus articles only (**T + A**). Results are shown in table 5.

	F1	Precision	Recall	Accuracy
<b>T</b>	.899	.826	<b>.985</b>	.895
<b>T + E</b>	.919	.891	.949	.921
<b>T + A</b>	.917	<b>.923</b>	.911	.922
<b>T + E + A</b>	<b>.927</b>	.915	.939	<b>.930</b>

Table 5: Ablation Results

From this table, we can see that using all three information sources performs best in terms of overall accuracy and F1 score. Although just using the tweet gives particularly high recall, the overall performance is significantly worse than the others, about 3% lower in accuracy and F1 compared with using all three. This shows strongly that GAPFORMER is leveraging the articles and entities effectively to achieve better accuracy.

## 5 Conclusions

We presented GAPFORMER, which detects informative Covid-19 tweets, using a graph classification system that leverages articles and entities. This is done by first building graphs using normalized pointwise mutual information to determine related nodes, and then combining pre-trained language models with GraphSAGE convolutions and a pooling mechanism to classify them.

As shown in the experiments, this way of incorporating additional data is effective. It enables this system to outperform all seven baselines tested.

In future work, we plan to extend and upgrade this system in several ways:

- Enable broader input data, e.g. tweet replies

and user information. This will include heterogeneous data rather than only text.

- Improve performance by collecting and using more labeled data.
- Adapt it to other tasks. For example, we believe GAPFORMER can be used to effectively detect Covid-19 misinformation.

We are also looking at real-world applications of this tool. Since accuracy and other metrics are already above 90%, we believe it can have practical value, especially with the improvements above. We aim therefore to find direct ways to use GAPFORMER to help mitigate the Covid-19 pandemic.

## References

- Piush Aggarwal. 2019. Classification approaches to identify informative tweets. In *Proceedings of the Student Research Workshop Associated with RANLP 2019*, pages 7–15. 1
- Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. 2019. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59. 2
- Firoj Alam, Fahim Dalvi, Shaden Shaar, Nadir Durrani, Hamdy Mubarak, Alex Nikolov, Giovanni Da San Martino, Ahmed Abdelali, Hassan Sajjad, Kareem Darwish, and Preslav Nakov. 2020. [Fighting the covid-19 infodemic in social media: A holistic perspective and a call to arms](#). 1
- Maged N Kamel Boulos and Estella M Geraghty. 2020. Geographical tracking and mapping of coronavirus disease covid-19/severe acute respiratory syndrome coronavirus 2 (sars-cov-2) epidemic and associated events around the world: how 21st century gis technologies are supporting the global fight against outbreaks and epidemics. 1
- Jeffrey Brainard. 2020. Scientists are drowning in covid-19 papers. can new tools keep them afloat. *Science*. 1
- Sijing Chen, Jin Mao, Gang Li, Chao Ma, and Yujie Cao. 2020. Uncovering sentiment and retweet patterns of disaster-related tweets from a spatiotemporal perspective—a case study of hurricane harvey. *Telematics and Informatics*, 47:101326. 1
- Limeng Cui and Dongwon Lee. 2020. [Coaid: Covid-19 healthcare misinformation dataset](#). 1
- Limeng Cui, Haeseung Seo, Maryam Tabar, Fenglong Ma, Suhang Wang, and Dongwon Lee. 2020. Deterrent: Knowledge guided graph attention network for detecting healthcare misinformation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 492–502. 2
- Thanh Vu Dat Quoc Nguyen and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English Tweets. *arXiv preprint*, arXiv:2005.10200. 2, 4
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. [BERT: pre-training of deep bidirectional transformers for language understanding](#). *CoRR*, abs/1810.04805. 1
- WA Falcon. 2019. Pytorch lightning. *GitHub*. Note: <https://github.com/PyTorchLightning/pytorch-lightning> Cited by, 3. 4
- Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, pages 1024–1034. 3
- Muhammad Imran, Prasenjit Mitra, and Carlos Castillo. 2016. Twitter as a lifeline: Human-annotated twitter corpora for nlp of crisis-related messages. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France. European Language Resources Association (ELRA). 1
- Nayomi Kankanamge, Tan Yigitcanlar, Ashantha Goonetilleke, and Md Kamruzzaman. 2020. Determining disaster severity through social media analysis: Testing the methodology with south east queensland flood tweets. *International journal of disaster risk reduction*, 42:101360. 1
- Shamant Kumar, Geoffrey Barbier, Mohammad Ali Abbasi, and Huan Liu. 2011. Tweetracker: An analysis tool for humanitarian and disaster relief. *ICwSM*, 11:78–82. 1
- L. Li, Q. Zhang, X. Wang, J. Zhang, T. Wang, T. Gao, W. Duan, K. K. Tsoi, and F. Wang. 2020. Characterizing the propagation of situational information in social media during covid-19 epidemic: A case study on weibo. *IEEE Transactions on Computational Social Systems*, 7(2):556–562. 1
- Ilya Loshchilov and Frank Hutter. 2017. [Decoupled weight decay regularization](#). 3, 4
- Zhibin Lu, Pan Du, and Jian-Yun Nie. 2020. Vgcnbert: Augmenting bert with graph embedding for text classification. In *European Conference on Information Retrieval*, pages 369–382. Springer. 2
- Lucas Ou-Yang. 2013. [Newspaper3k](#). 2

- 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. 2, 4
- 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*. 1, 3
- F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. 4
- Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. *Glove: Global vectors for word representation*. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. 3
- Umair Qazi, Muhammad Imran, and Ferda Ofi. 2020. Geocov19: a dataset of hundreds of millions of multilingual covid-19 tweets with location information. *SIGSPATIAL Special*, 12(1):6–15. 1
- Lei Qin, Qiang Sun, Yidan Wang, Ke-Fei Wu, Mingchih Chen, Ben-Chang Shia, and Szu-Yuan Wu. 2020. Prediction of number of cases of 2019 novel coronavirus (covid-19) using social media search index. *International Journal of Environmental Research and Public Health*, 17(7):2365. 1
- Corby Rosset, Chenyan Xiong, Minh Q. Phan, Xia Song, Paul. Bennett, and Saurabh Tiwary. 2020. Knowledge-aware language model pretraining. *ArXiv*, abs/2007.00655. 2
- Jyoti Prakash Singh, Yogesh K Dwivedi, Nripendra P Rana, Abhinav Kumar, and Kawaljeet Kaur Kapoor. 2019. Event classification and location prediction from tweets during disasters. *Annals of Operations Research*, 283(1):737–757. 1
- Bruno Takahashi, Edson C Tandoc Jr, and Christine Carmichael. 2015. Communicating on twitter during a disaster: An analysis of tweets during typhoon haiyan in the philippines. *Computers in human behavior*, 50:392–398. 1
- Hien To, Sumeet Agrawal, Seon Ho Kim, and Cyrus Shahabi. 2017. On identifying disaster-related tweets: Matching-based or learning-based? In *2017 IEEE Third International Conference on Multimedia Big Data (BigMM)*, pages 330–337. IEEE. 1
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008. 1
- WHO. 2020. World Health Organization coronavirus disease (covid-19) dashboard. [WHOCoronavirusDisease \(COVID-19\) Dashboard](https://www.who.int/emergencies/diseases/novel-coronavirus-2019/dashboard). Accessed: 2020-09-02. 1
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771. 4
- Ledell Wu, Fabio Petroni, Martin Josifoski, Sebastian Riedel, and Luke Zettlemoyer. 2019. Zero-shot entity linking with dense entity retrieval. In *arXiv:1911.03814*. 2
- Jiawei Zhang. 2020. G5: A universal graph-bert for graph-to-graph transfer and apocalypse learning. *arXiv preprint arXiv:2006.06183*. 2
- Chen Zhao, Chenyan Xiong, Corby Rosset, Xia Song, Paul N. Bennett, and Saurabh Tiwary. 2020. *Transformer-xh: Multi-evidence reasoning with extra hop attention*. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net. 2