Learning Domain-Specific Word Embeddings from COVID-19 Tweets

1st Steve Aibuedefe Aigbe
Department of Computer Science
University of Houston
Houston, USA
saaigbe@uh.edu

2nd Christoph Eick
Department of Computer Science
University of Houston
Houston, USA
CEick@central.uh.edu

2021 IEEE International Conference on Big Data (Big Data) | 978-1-6654-3902-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/BigData52589.2021.9671817

***Abstract*—The COVID-19 global pandemic has been a major catastrophic event that impacted the world's economy. During the pandemic there was a rise in the use of social media such as Twitter by people to express their reactions and responses to the global pandemic. This drove researchers to analyze these micro-blogging texts, using natural language processing (NLP) methods, to understand information inherent in those texts. Most of these NLP tasks employ the use of word embeddings in training neural network models. These word embeddings are mainly trained on general text corpus which produce sub-optimal performance when used in domain-specific NLP tasks such as in COVID-19 related tweets. In this paper, we present a learned COVID-19 tweets domain-specific word embeddings for use in COVID-19 related tweets NLP tasks. Our evaluation results show that our domain-specific COVID-19 tweets word embeddings perform better than pretrained general word embeddings in a downstream domain-specific NLP task. Our COVID-19 tweets word embeddings are available for use by researchers who wish to perform downstream NLP tasks with pretrained domain-specific COVID-19 tweets word embeddings.**

***Index Terms*—Domain-Specific, Word Embeddings, COVID-19, Tweets**

I. Introduction

Social media platforms such as Twitter have been used as media for people to express themselves when dealing with catastrophic events such as extreme political viewpoints [1], terrorism [2], natural disasters and global pandemic. The coronavirus disease of 2019 (COVID-19) global pandemic has been a major catastrophic event with severe impact on the world's economy, which led to rise in unemployment, and psychological and mental health issues. These sudden socioeconomic changes have motivated several researches where computational methods using machine and deep learning models have been prominent [3]–[5]. The rise in the use of social media such as Twitter during the COVID-19 pandemic drove researchers to analyze the micro-blogging texts, using natural language processing (NLP) methods, to understand people's reactions to the pandemic. Most of the NLP methods involves building neural network models that use word embeddings for several NLP tasks such as text classification, named entity recognition (NER) and part-of-speech (POS) tagging.

Word embedding is a technique in NLP that transforms the words in a text into dense vectors of real numbers in a continuous embedding space. While traditional NLP systems represent words as indices in a text, which do not capture the semantic relationships between words, word embeddings encode distributional semantics in learned and co-occurrence word vectors [6]–[8]. Appropriate word representations play a significant role in NLP, and this has led to the incorporation of word embeddings in a variety of NLP tasks such as sentiment analysis so as to boost their performance [3], [9]. While word embeddings have been demonstrated to be highly beneficial in these works, most of them are carried out with word embeddings generated from general-domain text corpus such as Wikipedia, general Twitter text and Google News, and their performances are sub-optimal when applied to specific-domain tasks. To achieve maximum benefits when using word embeddings for specific-domain NLP tasks, they need to be generated using the specific-domain text corpus.

In this paper, we propose and provide word embeddings trained on the specific domain of COVID-19 related tweets since several NLP works are being done on COVID-19 related tweets [3]–[5]. There are two implicit assumptions usually made about the effectiveness of word embeddings to down-stream NLP tasks. The first is that the training corpus for the word embeddings is available and much larger than the training data of the down-stream NLP task while the second is that the domain of the word embeddings corpus is closely aligned with the domain of the down-stream NLP task. Our word COVID-19 word embeddings satisfy these assumptions as the hundreds of millions of COVID-19 related tweets serve as a large enough corpus to train the word embeddings and its usage in NLP tasks for COVID-19 related tweets would provide better performance than general-domain word embeddings. We perform both intrinsic and down-stream extrinsic evaluation of our trained COVID-19 word embeddings to compare their performance with the general-domain word embeddings.

In this work, we propose to answer the following research questions:

- Is it possible to improve the performance of neural network models for NLP task if domain-specific word embeddings are used for training instead of word embeddings from general corpus?
- What hyperparameters, if any, influence the word embeddings performance? The hyperparameters for word

embeddings training include vector dimension, window size, min count
- Does the word embedding performance differ significantly from one deep learning model to another (CNN and BD-LSTM)?

In summary, the main contributions of our work include:

- We present a reference point for the comparison of using COVID-19 related tweets domain-specific word embeddings as against word embeddings produced from general-domain text corpus for COVID-19 related tweets NLP tasks.
- We provide COVID-19 related tweets word embeddings that can be used by other researchers who are interested in exploring NLP tasks on COVID-19 related Twitter dataset.

II. Related Work

Several works have been done on learning word embeddings [10]–[12]. More recently, word embeddings gained much popularity with the Word2Vec method proposed Mikolov et al [7]. This method includes two models: a continuous bag-of-words (CBOW) model and a skip-gram (Skip-Gram) model, and they both learn word embeddings from large-scale unsupervised text corpus. Many works have been done to improve word embeddings by extending the Word2Vec model such as the work of Levy and Goldberg [13] where they employed the syntactic contexts derived from automatically generated dependency parse-tree while the Word2Vec model uses a linear context.

Not very many studies have been done that focus on learning domain-specific word embeddings. Some of these studies have been done on sparse domain-specific datasets such as patents [14], radiological [15], and cybersecurity [16], [17]. All of these studies involve integrating the created word embeddings with pretrained general domain word embeddings to improve performance. Another significant work on domain-specific word embeddings is the work of Ghosh et al. [18] where they brought in-domain words close to each other in the embedding space while pushing out-domain words away from in-domain words. Our work involves using a very large COVID-19 related Twitter dataset to train COVID-19 related tweets domain-specific word embeddings without any interactions with general pretrained word embeddings.

Despite the prevalent use of word embeddings in natural language processing tasks, there has been no agreed best ways to evaluate the word embeddings. A variety of benchmarks, which are used widely to assess the quality of word embeddings, exist. These evaluation methods are largely classified into two categories: intrinsic evaluation and extrinsic evaluation. Intrinsic evaluation tends to quantify how well different kinds of linguistic structures can be detected with the word embeddings [19], [20]. For extrinsic evaluation, the performance of the word embeddings is assessed on downstream NLP tasks by measuring certain performance metrics peculiar to the tasks. In this paper we employ word semantic similarity and word analogy for the intrinsic evaluation of our trained word embeddings while we use sentiment analysis for downstream extrinsic evaluation of the word embeddings.

III. Word Embeddings Training

To train the COVID-19 word embeddings, we follow the steps as shown in Fig. 1.

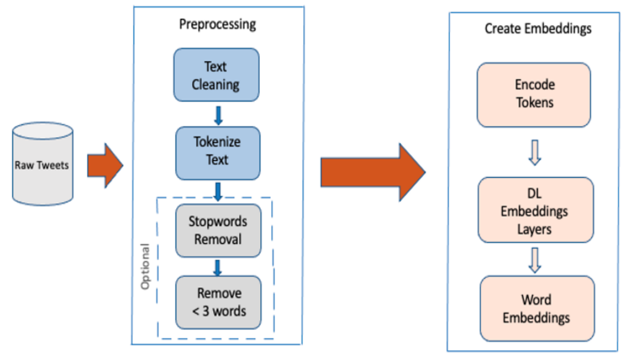

Fig. 1. Word Embeddings Creation Pipeline.

First, we extract the COVID-19 related tweets. Many extractions of COVID-19 related tweets have been done. However, due to Twitter's policy, tweets cannot be transferred or shared. Consequently, only the tweets IDs and dates are transferable. Thus, we need to get the tweets IDs and use them to extract the corresponding tweets details in a process termed hydration. The curated tweets IDs we used for this work was assessed from the dataset collected by Banda et al. [21] Tweets IDs can be hydrated using different available packages and applications such as Twarc and Hydrator via the Twitter API. The Twitter API has a limitation on the number of tweets that can be hydrated within a given duration, and the package that works best in managing this limitation is usually preferable for the hydrating task. Thus, the Hydrator application is used in this work.

A hydrated tweet ID has 34 attributes, which include tweet id, user id, created date, location, language, text, etc. We are only interested in the "text" field of the hydrated tweets. For this research work, we decided to extract tweets from March 2020 to July 2021. The reason behind this decision is to cover the whole COVID-19 pandemic milestones: from the beginning involving all the apprehensions, to the middle of the year when different control measures, such as lockdowns and restrictions, were introduced, and to more recently when vaccines were introduced and the people's hesitation to the vaccines. Over 100million English tweets were extracted for this research work. The basis of our research work is to have as many COVID-19 related tweets as possible since this is expected to generate a more robust word embedding, which we expect to give better performance on COVID-19 tweets related NLP task than word embedding created from general text corpus.

After the tweet hydration and extraction are complete, the extracted text field is preprocessed to remove the parts that do not contribute to the word embeddings. These parts include hashtags, URLs, retweet tags, mentions tags, special characters, numbers, gifs, emojis and emoticons. This preprocessing task is done on all the tweets and the filtered tweets are combined to form the sentences of the resulting large COVID-19 tweets corpus.

The resulting corpus of filtered tweet texts sentences are tokenized into words, which are vectorized using a one-hot encoding and thereafter fed to a neural network with a single hidden embedding layer. Table I shows the metadata of the COVID-19 word embeddings training dataset.

TABLE I
COVID-19 WORD EMBEDDINGS TRAINING DATASET

Metadata	Value
Number of tweets	100.1Million
Number of words in training	2.1Billion

The word embedding algorithm used is the Word2Vec (Skip-gram) algorithm. The training objective of the Skip-gram model is to find word embeddings that are useful for predicting the surrounding words in a sentence. Given a sequence of training words $W_1, W_2, W_3, \ldots, W_N$, the Skip-gram model aims to maximize the average log probability.

$$\frac{1}{N}\sum_{n=1}^{N}\sum_{-m\leq i\leq m, i\neq 0}\log p\left(W_{n+i} \mid W_n\right)$$

where m is the size of the training context. A larger m will result in more training data and can lead to a higher accuracy. The resultant learned weights of the neural network form the word embeddings. The word embeddings is generated as a key-value pair where the key is the word, and the value is the corresponding learned weight vector. The resulting word embeddings are saved to file either as a binary or text file, which can be loaded for use in any neural network model for conducting NLP tasks. The hyperparameters for training the word vectors include vector dimension, context window, epochs and minimum count. The vector dimension is the size of the vector representing a single word in the vocabulary used to train the word embeddings, and it is usually 25, 50, 100, 200, 300. We trained our word embeddings with vector dimensions of 100 and 300. The context window size is an integral part of the word embedding training algorithm as it helps to give semantic meaning to a word in a text, and it signifies the number of words before and after the target word of interest during training. We used a context window size of 10 in this work. The minimum count is the minimum frequency of words in the word embedding vocabulary, that is, words whose frequencies are lower than the minimum count will not be included in the word embeddings. We used a minimum count of 5 for training our word embeddings. Our word embeddings were trained for 20 epochs.

IV. WORD EMBEDDINGS EVALUATION

We evaluated the learned COVID-19 related tweets word embeddings using two intrinsic evaluation methods (word semantic similarity and word analogy) and one downstream extrinsic evaluation method (sentiment analysis). The word embeddings used for the evaluation are given in Table II.

TABLE II
WORD EMBEDDINGS FOR EVALUATION

Word Embeddings	Training Corpus
Glove.6B	Wikipedia English corpus
Glove.twitter.27B	General Twitter corpus
Google.negative300d	Google News
COVID-19	**COVID-19 Twitter corpus**

A. Intrinsic Evaluation

1) Word Semantic Similarity: This method is based on the idea that the distance between words in an embedding space could be evaluated via the human heuristic judgments on the actual semantic distances between these words. The assessor is given a set of pairs of words and asked to assess the degree of similarity for each pair. The distances between these pairs are also collected in a word embeddings space, and the two obtained distances sets are compared. We used the WordSim353 assessed word pairs for this evaluation. Here we load the pretrained word embeddings generated from Google News, Wikipedia, general tweets and our own learned COVID-19 tweets word embeddings. We then use the embedding matrices to compute similarities between the word pairs from WordSim353 assessed word pairs. The assessed word pairs have pairs of word with associated assessed similarity scores which serve as the ground truth for evaluating the similarity results from the word embeddings. For example, a sample of the assessed word pairs for WordSim353 is given below:

$computer \quad keyboard \quad 7.62$
$Jerusalem \quad Israel \quad 8.46$
$planet \quad galaxy \quad 8.11$
$canyon \quad landscape \quad 7.53$
$OPEC \quad country \quad 5.63$
$day \quad summer \quad 3.94$

The similarities between the pairs of words are determined by computing the cosine similarities between the pairs of words using their respective embedding vectors. A rank correlation is then computed between the similarities generated from the word embeddings and the ground truth. The closer the correlation is to 1.0 the better the word embeddings. Table III shows the correlation values for the pretrained word embeddings and our word embeddings on the WordSim353 word pairs.

2) Word Analogy: This method is also known as word semantic coherence, analogical reasoning, or linguistic regularities. It is based on the idea that arithmetic operations in a word vector space could be predicted by humans. For example, given a set of three words, a, $a*$ and b, the task is to identify such word $b*$ that the relation b:$b*$ is the same as

TABLE III
WORD SEMANTIC SIMILARITY EVALUATION RESULTS

Word Embeddings	Correlation Value (WordSim353)
Glove.6B	0.658
Glove.twitter.27B	0.378
Google.negative300d	0.629
COVID-19	**0.626**

the relation a:$a*$. For instance, one has words $a = Paris$, $b = France$, $c = Moscow$. Then the target word would be $Russia$ since the relation $a : b$ is $capital : country$, so one needs need to find the capital of which country is Moscow. We used the Google Analogy Test Set for the word analogy evaluation, and a sample from this test set is given below:

$dublin \quad ireland \quad kathmandu \quad nepal$
$lusaka \quad zambia \quad tehran \quad iran$
$rome \quad italy \quad windhoek \quad namibia$
$comfortable \quad uncomfortable \quad clear \quad unclear$
$good \quad better \quad high \quad higher$
$finger \quad fingers \quad onion \quad onions$
$play \quad plays \quad sing \quad sings$

Each of the word embeddings is evaluated against this test set where it is expected to give the correct 4$^{\text{th}}$ word in each set. This task involved feeding a prediction model, created with each of the word embeddings, the first 3 words of each line of the test set and the model is expected to correctly predict the 4$^{\text{th}}$ word in each line of the test set. The higher the accuracy of generating the 4$^{\text{th}}$ word the better the word embeddings. Table IV shows accuracies of the Google Analogy Test set for each of the word embeddings:

TABLE IV
WORD ANALOGY EVALUATION RESULTS

Word Embeddings	Accuracy Value (Google Analogy)
Glove.6B	0.809
Glove.twitter.27B	0.536
Google.negative300d	0.728
COVID-19	**0.603**

B. Extrinsic Evaluation - Sentiment Analysis

To test the performance of the learned COVID-19 related tweets word embedding against the other frequently used word embeddings for downstream NLP tasks we used the COVID-19 sentiment analysis task. This assumption this research tests is that the accuracies for COVID-19 related tweets word embeddings should be better than those of the other word embedding for the sentiment analysis of COVID-19 related tweets. Sentiment analysis is an NLP text classification task that classify texts into different sentiments such as negative, neutral, and positive. For our sentiment analysis testing we will used the dataset of COVID-19 related tweets already labeled as either negative, neutral, or positive sentiment. This research work requires COVID-19 related tweets as the premise of our research is based on the using COVID-19 word embeddings for NLP tasks on COVID-19 related tweets. The sentiment labeled COVID-19 related twitter dataset used for this research is the dataset provided by Gupta et al. [22], and 400K of these dataset was used for the evaluation.

We perform the same preprocessing tasks done while generating the word embeddings on the dataset to prepare it for our deep learning models. We tested with two deep learning models: BD-LSTM (Bidirectional Long Short Term Memory) and CNN (Convolution Neural Network). For both models we use 80% of the dataset for training and the remaining 20% for validation, and the validation accuracies are reported. The BD-LSTM model comprises an embedding layer as the input layer, 2 bidirectional LSTM and 1 dense layer as the hidden layers, and another dense layer as the output layer. On the other hand, the CNN model is a deep neural model with four hidden layers (3 Conv1D layers & 1 dense layer), a dense output layer and an embedding layer serving as the input layer of the model. The hidden layers have filters as 128 with kernel size of 5. We run each of the models for 20 epochs using both lower and higher dimension vector sizes.

TABLE V
SENTIMENT ANALYSIS FOR COVID-19 RELATED TWEETS - LOWER DIMENSION VECTORS

Word Embeddings	BD-LSTM Accuracy	CNN Accuracy
Glove.6B.100d	90.07%	86.94%
Glove.twitter.27B.100d	89.92%	87.66%
COVID-19.100d	**90.24%**	**87.37%**

TABLE VI
SENTIMENT ANALYSIS FOR COVID-19 RELATED TWEETS - HIGHER DIMENSION VECTORS

Word Embeddings	BD-LSTM Accuracy	CNN Accuracy
Glove.6B.300d	90.33%	87.82%
Glove.twitter.27B.200d	90.31%	88.24%
Google.negative300d	90.60%	88.98%
COVID-19.300d	**90.74%**	**89.17%**

V. DISCUSSION

Analysis the results of the intrinsic evaluation of our COVID-19 tweets word embeddings in tables III and IV, we can see that our word embeddings returns similar values compared to the commonly used pretrained word embeddings for downstream NLP tasks. However, it is noted in literature that results of the intrinsic evaluation of word embeddings are not an indicator of poor or good performance when the word embeddings are used in downstream NLP tasks [23], [24].

In tables V and VI, the accuracies for the sentiment analysis using the different word embedding are recorded for each of the deep learning models (BD-LSTM & CNN). Each of the model is trained and tested with sentiment labeled COVID-19 related twitter dataset. Table Vcontains results for using low dimensional word embeddings while table VIcontains results for high dimensional word embeddings.

From the BD-LSTM model, we notice the following observations. The higher dimension word embeddings performed better than the lower dimension word embeddings on the sentiment analysis test for all the word embeddings. The COVID-19 word embedding has better performance than the other word embeddings both in lower and higher dimensions.

From the CNN model, we notice the following observations. All the word embeddings performed better in higher dimension than in lower dimension. While the COVID-19 word embeddings return the best performance in higher dimension, it was out-performed by the glove.twitter.27B word embeddings in lower dimension.

From both tables V and VI, we can see that all the word embeddings perform better with the BD-LSTM model than with the CNN model in both lower and higher dimensions. Consequently, we can conclude that a BD-LSTM model is better for sentiment analysis than a CNN model. Also, from both tables, we can conclude that higher dimensional word embeddings perform better than lower dimension word embeddings.

VI. COVID-19 Tweets Word Embeddings Dataset

The learned COVID-19 tweets word embeddings from the research work is made available in the following GitHub repository *(https://github.com/steveaigbe/COVID_19_Word_Vectors)*. Researchers can find details of how to assess and use the word embeddings in the repository.

VII. Conclusion and Further Works

In this paper we present the learning of domain-specific word embeddings form COVID-19 related Twitter dataset. We evaluated the learned COVID-19 word embeddings on both intrinsic and extrinsic evaluation methods. Our results show that our COVID-19 tweets word embeddings perform better than other commonly used pretrained word embeddings for domain-specific NLP tasks. Also, our result show that word embeddings with higher dimension vector size perform better than those with lower dimension. Finally, we can conclude that our COVID-19 word embeddings perform better than all the other word embeddings when used in an LSTM neural network model for downstream NLP tasks. Although our word embedding is trained on a relatively smaller corpus compared to the other pretrained word embeddings, yet it produces better performance. This reinforces the assertion that domain-specific word vectors perform better for NLP tasks conducted within that domain.

The COVID-19 tweets word embedding provided with this paper can be used by an researcher who wishes to perform any research work that involves carrying out NLP task on COVID-19 related tweets using word embeddings. Those researchers need not re-invent the wheel by learning new COVID-19 tweets domain-specific word embeddings, they can apply our word embeddings in their works. Also, Our word embeddings is relatively smaller than the available pretrained word embeddings, and this will present the opportunity for less computational resources for researchers who wants to use the word embeddings.

For further works, we plan to use the learned COVID-19 tweets domain-specific word embeddings to develop a language model for the detection of depression, stress and anxiety in COVID-19 related tweets.

Acknowledgment

This work was completed in part with resources provided by the Research Computing Data Core at the University of Houston.

References

[1] M. Alizadeh, I. Weber, C. Cioffi, S. Fortunato, and M. Macy, "Psychology and morality of political extremists: evidence from twitter language analysis of alt-right and antifa," *EPJ Data Science*, vol. 8, 12 2019.

[2] P. Garg, H. Garg, and V. Ranga, "Sentiment analysis of the uri terror attack using twitter," *2017 International Conference on Computing, Communication and Automation (ICCCA)*, pp. 17–20, 2017.

[3] R. Chandra and A. Krishna, "COVID-19 sentiment analysis via deep learning during the rise of novel cases," *CoRR*, vol. abs/2104.10662, 2021.

[4] M. Müller, M. Salathé, and P. E. Kummervold, "Covid-twitter-bert: A natural language processing model to analyse COVID-19 content on twitter," *CoRR*, vol. abs/2005.07503, 2020. [Online]. Available: https://arxiv.org/abs/2005.07503

[5] K. Glandt, S. Khanal, Y. Li, D. Caragea, and C. Caragea, "Stance detection in COVID-19 tweets," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Aug. 2021, pp. 1596–1611.

[6] T. Mikolov, W.-t. Yih, and G. Zweig, "Linguistic regularities in continuous space word representations," in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Jun. 2013, pp. 746–751.

[7] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013.

[8] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Oct. 2014, pp. 1532–1543. [Online]. Available: https://aclanthology.org/D14-1162

[9] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for Twitter sentiment classification," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jun. 2014, pp. 1555–1565.

[10] A. Mnih and G. Hinton, "Three new graphical models for statistical language modelling," in *Proceedings of the 24th International Conference on Machine Learning*, ser. ICML '07. Association for Computing Machinery, 2007, p. 641–648.

[11] A. Mnih and G. E. Hinton, "A scalable hierarchical distributed language model," in *Advances in Neural Information Processing Systems*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds., vol. 21. Curran Associates, Inc., 2009.

[12] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th International Conference on Machine Learning*, ser. ICML '08. Association for Computing Machinery, 2008, p. 160–167.

[13] O. Levy and Y. Goldberg, "Dependency-based word embeddings," in *ACL*, 2014.

[14] J. Risch and R. Krestel, "Domain-specific word embeddings for patent classification," *Data Technologies and Applications*, vol. 53, no. 1, pp. 108 – 122, 2019.

[15] T. L. Chen, M. Emerling, G. R. Chaudhari, Y. R. Chillakuru, Y. Seo, T. H. Vu, and J. H. Sohn, “Domain specific word embeddings for natural language processing in radiology,” *Journal of Biomedical Informatics*, vol. 113, p. 103665, 2021.

[16] A. Roy, Y. Park, and S. Pan, “Learning domain-specific word embeddings from sparse cybersecurity texts,” *CoRR*, vol. abs/1709.07470, 2017.

[17] R. Arpita, P. Youngja, and P. Shimei, “Incorporating domain knowledge in learning word embedding,” in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, 2019, pp. 1568–1573.

[18] S. Ghosh, P. Chakraborty, E. Cohn, J. S. Brownstein, and N. Ramakrishnan, “Designing domain specific word embeddings: Applications to disease surveillance,” *CoRR*, vol. abs/1603.00106, 2016.

[19] M. Baroni, G. Dinu, and G. Kruszewski, “Don’t count, predict! a systematic comparison of context-counting vs. context-predicting semantic vectors,” in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Jun. 2014, pp. 238–247.

[20] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, “Evaluation methods for unsupervised word embeddings,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, Sep. 2015, pp. 298–307.

[21] J. M. Banda, R. Tekumalla, G. Wang, J. Yu, T. Liu, Y. Ding, K. Artemova, E. Tutubalina, and G. Chowell, “A large-scale COVID-19 Twitter chatter dataset for open scientific research - an international collaboration,” May 2020. [Online]. Available: https://doi.org/10.5281/zenodo.3870894

[22] R. Gupta, A. Vishwanath, and Y. Yang, “Global Reactions to COVID-19 on Twitter: A Labelled Dataset with Latent Topic, Sentiment and Emotion Attributes,” August 2021. [Online]. Available: https://doi.org/10.3886/E120321V10

[23] A. Bakarov, “A survey of word embeddings evaluation methods,” *CoRR*, vol. abs/1801.09536, 2018. [Online]. Available: http://arxiv.org/abs/1801.09536

[24] T. Schnabel, I. Labutov, D. Mimno, and T. Joachims, “Evaluation methods for unsupervised word embeddings,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Sep. 2015, pp. 298–307.