

# A Computational Analysis of Climate Change Sentiment on Social Media

Shon Verch

## Abstract

Despite overwhelming scientific consensus regarding climate change, the basis for climate change still remains a wildly debated and political issue. In this paper, we analyse a dataset of over 600M English-language tweets to understand how the climate change sentiment has changed over time. We train a series of word2vec models, one per month, to examine how words used in the context of climate change have changed over time. We also fine-tune a BERT model to perform sentiment classification on tweets.

## 1 Introduction

The United States is the second largest emitter of greenhouse gasses, only behind China; in 2018, the United States emitted an estimated 5.4 billion metric tons of CO<sub>2</sub>, 15% of the total CO<sub>2</sub> emissions in that year (36.58 billion metric tons) [2, 34, 40]. But, despite this, the basis for climate change remains a wildly debated and political issue; the strategies for tackling climate change in the United States have wildly varied between Presidential administrations. While the Obama administration enacted new climate change policy, the Trump administration has simultaneously rolled them back [17]. This huge disparity between strategies and overall attitude towards climate change between administrations suggests a dichotomy in climate change sentiment among the general public—or at least the voting populous.

One such reason for the social and political divisiveness regarding climate change might have to do with the fact that climate change is still a relatively new issue in the mainstream. While studied for decades by academics, research about the impacts of climate change was largely ignored by the general public, and in large part due to a lack of awareness on the issue; however, in recent decades, climate change has become one of the world’s foremost concerns, and ever so prevalent in mainstream media [11, 25]. In the last decade alone, the percentage of American adult voters that acknowledge that climate change is happening increased by 13%, meanwhile the percentage that said that “most scientists think that global-warming is happening” increased by 20%, Figure 1 depicts the climate views of Americans over a period of ten years. Similarly, climate change has become more widely understood as a serious threat to our species and planet by the American public. Between 2008 and 2018, the percentage of Americans that said that climate change will harm future generations, plants and animals, developing

countries, and the United States, increased by nearly 15%. Figure 2 shows the risk perception of climate change over a period of ten years. And this trend does not exclusively apply to the United States either. Climate change awareness has increased dramatically within the last decade *across the board*. Since at least 2006, concerns about climate change have increased internationally in countries such as France, Canada, and the United Kingdom, which have seen a more than 10% increase in the percentage of people saying that climate change is a major threat. Figure 4 shows how the international concern about climate change has been increasing in the last decade.

Yet, despite this seemingly universal shift in attitudes regarding climate change, and increasing awareness at a global level, there still exists a deep-rooted political divide in climate change views. At a national level, the majority of Americans agree that the climate change is a major concern. However, at the state and county level, differences in climate change attitudes across voters tend to be driven by political views. In historically Republican states (based on past elections) such as North Dakota, Oklahoma, Alabama, and Kentucky, the percentage of adults who indicated that they were worried about global warming in 2020 was roughly 10% *below* the national average (63%). Whereas, in historically Democratic states such as New York, New Jersey, California, and Maryland, the percentage of adults who said that they were worried about global warming was roughly 8% *above* the national average [9, 26].

Indeed, since 2013, those who identify as Democrats have been substantially more likely than those who identify as Republicans to report concern regarding climate change. In a 2018 survey conducted by the Pew Research Center, roughly a quarter (27%) of Republicans said that climate change is a major threat compared with more than three-quarters (83%) of Democrats [19]. Figure 3 shows the climate views of Americans based on political associations. Notice that respondents that identified as Conservative Republicans were markedly less likely to indicate that they believe that global warming is happening, while those who identified as Liberal Democrats were overwhelmingly likely to indicate belief. However, for both partisan groups, there has been little change in climate belief over time: Liberal Democrats have consistently indicated a strong belief, while Conservative Republicans have generally indicated weak belief that global warming is happening.

This all to suggest that political views are inextricably linked with climate change beliefs and attitudes. The significant divide between climate change concerns among Liberals and Conservatives reveals the political nature of the issue. While this is not too surprising since climate change often manifests itself as

such, given that the most basic fact of climate change—whether or not it is even happening—is still widely disagreed upon by two polar opposites on the political spectrum, it is clear that the views that underpin climate change beliefs are strongly linked with political attitudes. And at the same time, political views are strongly shaped and influenced by the news, discourse, and media around us. This is especially true on platforms such as Twitter and Facebook, where online political activities have been shown to strongly correlate to political awareness, activity, and participation in the real-world [3, 6]. In particular, the explosion in climate change awareness in the last two decades has occurred in tandem with the advent and mass-adoption of social media. The Web has enabled real-time communication across the globe, and as a result, most news consumption has shifted from physical forms to online social media, as it is more convenient, accessible and easier to share than physical news [22, 32]. Producing and publishing content is now easier than ever. What may have required an industry-backed media publisher twenty years ago, can now be done by a single person, independent of any large organisation. But, this low barrier to entry can be a doubled-edged sword: while it makes it easy and affordable for budding journalists and other content creators to get started, it also makes it extremely easy for malicious actors to disseminate low-quality and false information—and at an unprecedented rate. Vosoughi et. al. found that false news and misinformation spread significantly faster, farther, and deeper than the truth on Twitter, and the effects were more pronounced for false political news than other categories (terrorism, science, disasters, and finance) [49].

Mass discussion, discourse, and reporting about climate change on social media has inevitably created a broad spectrum of opinions regarding the issue [18]. In this respect, while climate change awareness, acceptance, and activism has grown exponentially in recent decades, this has also had the effect of amplifying the reach of misleading information.

Reporting about climate change in the mainstream is often rife with misinformation such as clickbait, journalistic bias, and sensationalization [14]. Samantray and Pin noted that “from empirical results [by applying VADER sentiment analysis on tweets] that anti-climate change tweets are largely not credible,” which suggests that the spread of misinformation regarding climate change in social networks is largely due to bias and sensationalization [35]. Similarly, in a review on the effects of social media use on climate change opinion, knowledge, and behaviour, Anderson noted that while early research suggests social media has had a generally positive impact on climate change opinion, knowledge, and behaviour, social media does also “provide space for framing climate change skeptically and activating those with a skeptical perspective of climate change” [5].

With the meteoric rise in adoption of social media platforms as news outlets, especially among youth, along with the spread in fake news and misinformation on these same platforms, we desire further data and analysis on climate change sentiment on social media platforms such as Twitter to better understand the effect and efficiency of fake news [21, 37].

In this paper, we focus our attention on Twitter, and perform

a temporal analysis of climate change sentiment and beliefs on tweets over a span of three years. In specific, we analyse a dataset of over 600M English-language tweets to answer the following research questions:

1. **RQ1: How have attitudes on climate change and related keywords changed over time on Twitter?**
2. **RQ2: In what context are words used in discussions referencing climate change? Is there a political element?**
3. **RQ3: How has misinformation and fake news impacted climate change beliefs on Twitter?**

We take a data-driven three-pronged approach to analysing the tweets. Following the methodology from Schild et. al. and Zannettou et. al., we use the word2vec language model architecture to learn a set of word embeddings on tweets, in order to analyse their contents [36, 56]. Next, we train a text classifier to predict the climate change sentiment on a dataset of 50K tweets, which we then extrapolate on unseen data to predict the sentiment of new tweets.

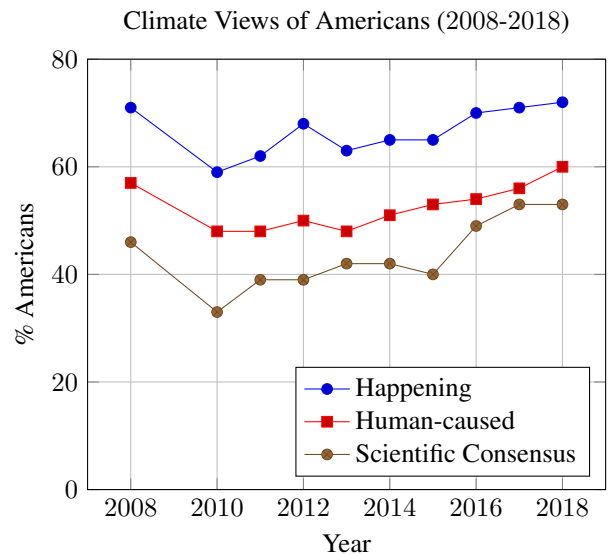


Figure 1: The climate beliefs of adult American voters from 2008 to 2018 [9, 26]. The graph shows the percentage of respondents who gave an affirmative answer the following questions: “Do you think that global warming is happening?” (Happening), “Assuming global warming is happening, do you think that it is caused mostly by humans” (Human-caused), and “Most scientists think global warming is happening?” (Scientific Consensus).

## 2 Datasets

To study climate change sentiment on social media, we collect four large-scale datasets of tweets for different stages of the analysis. In this section, we first provide a brief summary of

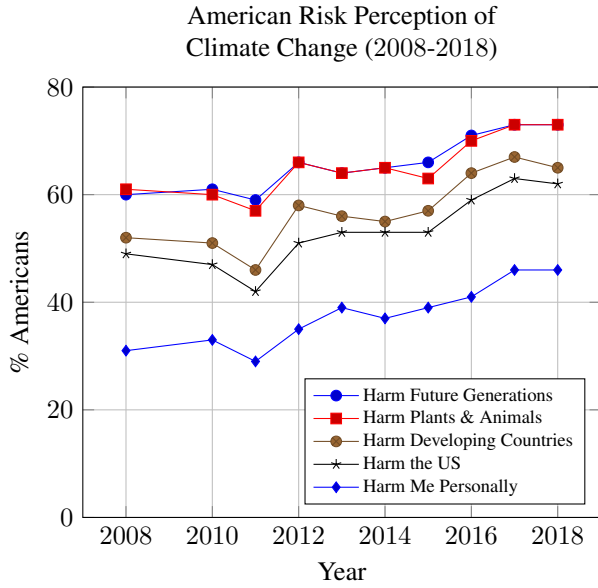


Figure 2: The risk perception of climate change among adult American voters from 2008 to 2018 [9, 26]. The graph shows the percentage of respondents who gave an affirmative answer to the following statement: “Global warming will harm future generations”, “Global warming will harm plants and animals”, “Global warming will harm developing countries”, “Global warming will harm the US”, and “Global warming will harm me personally.”

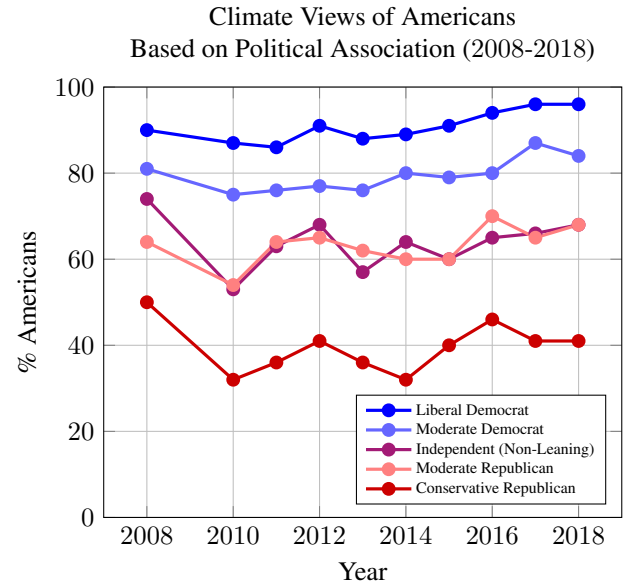


Figure 3: The climate views of adult American voters from 2008 to 2018 based on political association (the partisan group the respondent identifies with) [9, 26]. The graph shows the percentage of respondents who said that they believe that global warming is happening, grouped by political association.

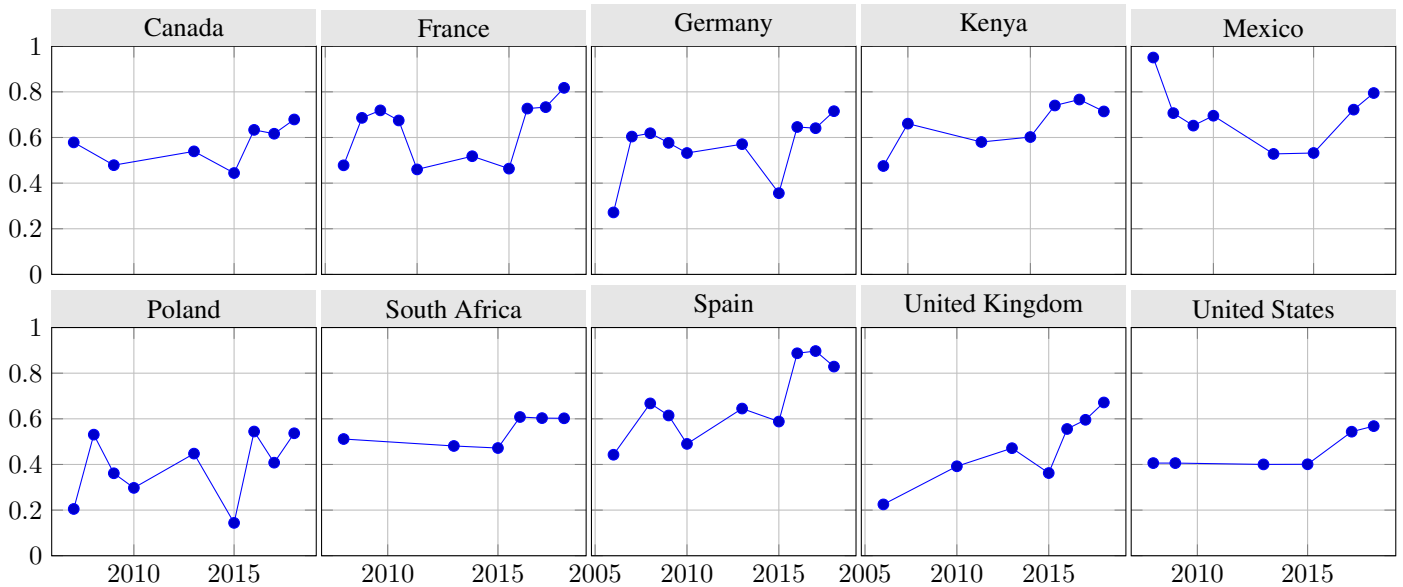


Figure 4: The proportion of respondents from each country who indicated that they have major concerns about climate change from 2006 to 2018. The countries shown above saw an increase of *at least* 10% in this time period. Aggregated from the Pew Research Center *Global Attitudes & Trends Surveys* between 2006 and 2018 [12].

each dataset. Then, we introduce how we process and clean the data so that it can be used for analysis.

**Twitter.** We focus on Twitter, an extremely popular mainstream microblogging and social networking service used by hundreds of millions of users [28]. On Twitter, people can post and interact with others in messages up to 280 characters long [45]. The relatively small size of each message makes it ideal for large-scale data mining and processing.

A monthly archive of tweets provided by the *Archive Team* were accessed from the *The Internet Archive* [7]. The archives are obtained from Twitter using the Streaming API, which provides a 1% random sample of all tweets made available on the platform, in real-time [46]. The archives are provided in monthly installments from February 2011 to June 2020. We collected tweets posted between January 1, 2018 and June 31, 2020, and then filtered each monthly archive for only tweets in English and applied data cleaning techniques. Section 2.1 outlines the techniques used to clean the tweets. In total, after filtering and cleaning the data, we were left with 582,395,451 tweets.

**Sentiment Classification.** To determine the climate change sentiment expressed in a given tweet, we use the Twitter Climate Change Sentiment Dataset provided by Edward Qian on *Kaggle* [33]. The dataset contains 43,943 tweets pertaining to climate change collected between April 27, 2015 and February 21, 2018. Each tweet was labelled by three independent reviewers, and only tweets with unanimous labels were kept [33]. Each tweet is labelled as one of the following classes:

- News (2): factual news about climate change.
- Pro (1): positive stance belief of man-made climate change.
- Neutral (0): neutral stance on the belief of man-made climate change.
- Anti (-1): negative stance on the belief of man-made climate change.

The data is provided as a CSV (comma-separated values) file with three columns: the sentiment class of the tweet (sentiment), the body of the tweet (message), and the unique Twitter id of the tweet (tweet id). For our analysis, we only use the the first two columns: sentiment and message.

## 2.1 Cleaning Twitter Data

Each monthly archive takes about 40–50 GB in storage space (compressed), and the total archives are estimated to be about 4.8 TB in size (compressed). This is due in large part to the fact that the raw dataset contains metadata for each tweet. To reduce the size of the dataset, we perform multiple cleaning steps. Each monthly archive of data is organised in a highly granulated folder structure. A snapshot of 1% of all Tweets on the platform is provided for each minute, of every hour, of every day in the month as distinct JSON files. For each of these JSON files, we apply a set of preprocessing operations.

First, we filter each monthly archive for only tweets in English and also discard irrelevant metadata (see the `clean_twitter_data.py` script). We only keep the following variables:

- `created_at`: the time the tweet was created.
- `id`: the id of the tweet as assigned by Twitter.
- `truncated`: boolean indicating whether the text was truncated.
- `coordinates` (if not null): location of the client as a geo-JSON object.
- `place` (if not null): a `Place` object where the tweet is associated.
- `quote_count` (if not null or 0): approximately how many times the tweet has been quoted.
- `reply_count` (if not null or 0): number of times the tweet has been replied to.
- `retweet_count` (if not null or 0): number of times the tweet has been retweeted.
- `favorite_count` (if not null or 0): approximately how many times the tweet has been linked.
- `lang`: the BCP 47 language id as corresponding the machine-detected language of the tweet, or und if no language could be detected.

Stripping out unnecessary metadata significantly reduces the file size of the dataset into manageable chunks. Table 1 provides a detailed summary of the size of the data before and after cleaning.

Next, from the JSON twitter data, we construct a corpus for training the word2vec models (see the `make_twitter_corpus.py` script). As our goal is to learn word embeddings, we simplify the data by removing Unicode characters, links, and Twitter handles (strings of the form “@username”).

To reduce the size of the vocabulary and group up words, we limit consecutive character repetitions to a maximum of three characters. So, for example, “Maaaaaario” becomes “Maaario”. While we do not want to absolutely eliminate words with repeated characters, since there is meaning in repeating characters consecutively (for example, to indicate excitement), the intuition behind this approach is that little semantic information is gained from repeated characters beyond a certain threshold. By grouping words such as “Hiii” and “Hiiiiiii” into a common group (“Hiii”) we are able to preserve the meaning, while simplifying and reducing the size of the dataset.

Finally, we use a rule-based method for expanding contractions (using the `contractions` Python package). Expanding contractions leaves the meaning of the sentences intact while simplifying the vocabulary, making it easier to train our models.

Archive Date (yyyy/mm)	Raw Size (GB)	Cleaned Size (GB)	Total Tweets (cleaned)
2018/02	3	0.405	186149
2018/03	25	0.401	1495995
2018/04	72	1	4332591
2018/05	97	1	5742705
2018/06	271	4	16548671
2018/07	354	8	24750548
2018/08	489	7	29084235
2018/09	502	7	30150913
2018/10	494	7	29252279
2018/11	479	7	28484241
2019/01	227	3	13162722
2019/02	390	5	22525520
2019/03	382	5	21751923
2019/04	463	7	26877282
2019/05	506	7	28897295
2019/06	460	6	25972116
2019/07	554	7	31101620
2019/08	649	6	22649080
2019/09	418	6	23610039
2020/01	616	7	35326304
2020/02	15	0.203	923193
2020/03	596	7	31649220
2020/04	721	8	40830461
2020/05	722	7	41516678
2020/06	733	9	45573671
<b>Total</b>	10,238	132.654	582,395,451

Table 1: A detailed summary of the size of the data before and after cleaning. After cleaning the data, we were able to reduce the size by approximately 7700%.

### 3 Methodology

In this section, we present the necessary background information and methodology used for the analysis of tweets. Then, we explore various visualisation techniques for understanding high-dimensional features. We also provide a framework for reproducing all experiments and visualisations in this paper, and give a brief summary of how it works.

#### 3.1 Representing Words as Vector Spaces

To analyse the content of a tweet, we first need to understand the meanings of the words that make up that tweet. If we can understand how words used on Twitter relate with one another, then we can begin to analyse and interpret semantic information about tweets as a whole. However, words in their human-friendly string representation are not particularly useful for analysis—there just isn’t much we can do with a strings. To perform useful computation, we need to first transform words into numbers. We need an *encoding scheme*.

**One-hot Encodings.** One of the simplest ways we can repre-

Vocabulary	Mango	Kiwi	Lime	Tomato
Mango	1	0	0	0
Kiwi	0	1	0	0
Lime	0	0	1	0
Tomato	0	0	0	1

Figure 5: An example of one-hot encoded vectors. Each vector has a dimension for each distinct word of the vocabulary, and each component is a binary value.

sent words as numerical values is using *one-hot encoding*. For some vocabulary  $V$  and word  $w \in V$ , the one-hot encoding of  $w$  is a vector with  $|V|$  dimensions (one for each distinct word in the vocabulary) where the  $i$ -th dimension is a binary value indicating whether  $w$  is the word represented by the  $i$ -th dimension. Figure 5 illustrates a few words encoded as one-hot vectors.

One-hot encodings are a popular choice when encoding categorical variables as they are fairly simple structures and can be constructed quickly and easily. However, their use is very impractical with large vocabularies, as the sparse representation demands large amounts of memory. For a vocabulary of size  $n$ , encoding all words has a space complexity of  $\Theta(n^2)$ . This makes it computationally infeasible to use on large datasets (where the vocabulary can easily exceed 100K words). Furthermore, one-hot vectors do not contain any inherent meaning—the construction of the one-hot vectors only depends on the position of words in the vocabulary. Swapping the order of two words in the vocabulary will change their one-hot vectors. This makes it impossible to reason about word meaning with one-hot vectors.

**Word Embeddings.** We desire a scheme for encoding words into vectors that is both low-dimensional (space efficient) and which encodes meaning. One reasonable metric for a ‘good’ encoding model is to say that words which are more similar should be closer together in the vector space. Notice how one-hot encoding does not satisfy this condition.<sup>1</sup>

A word embedding model,  $\mathcal{W}$ , is an  $n$ -dimensional real vector space, which maps words from a vocabulary,  $V$ , to an element of the vector space:  $\mathcal{W} : V \rightarrow \mathbb{R}^n$ . This vector space has the property that it *encodes* rich semantic information. If well trained, one excellent consequence of word embeddings are that we can manipulating points in the vector space to reveal language features such as *analogies* and *word similarity*. For example, assume  $\mathcal{W}$  is well-trained on the English language, then  $\mathcal{W}(\text{king}) - \mathcal{W}(\text{man}) + \mathcal{W}(\text{woman}) \approx \mathcal{W}(\text{queen})$ , where equality is measure of vector similarity.

Word2Vec is one approach to modelling words as dense vectors in a high-dimensional vector space introduced by Mikolov et. al. [27]. In this way, we *embed* words into a dense vector space. More specifically, we *learn* these embeddings with a shallow neural-network, in the hopes that we these embed-

<sup>1</sup>What does it even mean to take the distance or similarity between two one-hot vectors?

dings can approximate word similarity and reveal hidden relationships (such as analogies) between words.

**The Skip-Gram Model.** There are multiple techniques for training a word2vec model; however, the general approach is to train the network to solve a language-related task on one-hot encoded words, and in the meantime let it learn an internal word embedding representation. Then, we simply extract the learned latent space to obtain our word embeddings. As a consequence, the choice for the task which our network ‘solves’ is a critical decider of performance.

One approach which has proven to be very effective is the skip-gram architecture: a shallow neural-network trained to predict the context (or neighbourhood) of a target word [20, 27]. The words in this neighbourhood are called *context words* because the intuition is that neighbouring words are contextually meaningful.

Notice that while the skip-gram model is a supervised learning algorithm, we can generate training data in an unsupervised manner. Given some text document,  $\mathcal{D}$ , which represents a sequence of words, a sliding window of size  $d$  is used to construct sample pairs. For some word at index  $i$ , at most  $2d$  training samples are constructed by taking the following pairs of words:

$$(\mathcal{D}_i, \mathcal{D}_{i-d}), (\mathcal{D}_i, \mathcal{D}_{i-d+1}), \dots, (\mathcal{D}_i, \mathcal{D}_{i+d}).$$

Figure 6 illustrates how these target-context word pairs are constructed.

The skip-gram model consists of three layers: an input layer with  $|V|$  neurons, a hidden layer with  $N \leq |V|$  neurons, and an output layer with  $|V|$  [27]. The model takes in a one-hot encoded vector  $\mathbf{X}$  as input and outputs a probability vector  $\mathbf{Y}$  representing the probability distribution of the context over all words in the vocabulary. We use the softmax activation function to produce to normalize the outputs of the model to a probability distribution. The hidden layer contains a learned representation of the words. The weights of this layer,  $\mathbf{W}^{|V| \times N}$ , which we call the *projection weights*, is a matrix representing the latent vector space containing our word embeddings: the  $i$ -th row of  $\mathbf{W}$  gives the embedding vector of word  $i$ . Figure 7 illustrates this architecture.

**Subsampling.** The skip-gram approach of predicting the context word essentially boils down to a classification problem. In these sorts of problems, we want our data to be balanced so that it is representative of the true probability distribution. By generating training data using the sliding window approach, we oversample words which appear more often in the corpus. To remedy this, we use a technique proposed by Mikolov et. al. where we sample words according to a probability distribution designed to favour less frequently occurring words. This main advantage of subsampling is that we reduce the number of training examples, while maintaining the quality of the model. The sampling probability for a word  $w \in V$  is given by

$$P(w) = \min \left\{ \frac{t}{f(w)} \sqrt{\frac{f(w)}{t} + 1}, 1 \right\},$$

where  $f : V \rightarrow \mathbb{Z}$  gives the relative frequency of  $w$  in the corpus [27].

Sample Construction via Sliding Window ( $d = 2$ )	
Sentence	Contexts
[Sometimes we have] to fail to succeed.	we have sometimes
[Sometimes we have to] fail to succeed.	have to sometimes
[Sometimes we have to fail] to succeed.	we to fail
...	...
Sometimes we have to [fail to succeed] .	fail to

Figure 6: Example target-context pairs for the sentence, “Sometimes we have to fail to succeed”. The radius of the sliding window is 2 words.

**Training.** We train the model with a process called *stochastic gradient descent* (SGD). A computational approach to minimizing a function. Training consists of the following steps:

1. Generate a target-context pair,  $(w_t, w_c)$ , mapping a target word,  $w_t$ , to one of its neighbours,  $w_c$  (within a predefined window size).
2. Forward propagate the one-hot vector representation of the input word,  $w_t$ , through the model.
3. Calculate the training error according to a loss function,  $\mathcal{L}$  (we use a variant of the cross entropy loss [20, 27]).
4. Perform propagation to compute the gradient of the loss function with respect to the weights.
5. Update the weights:  $\mathbf{W}_t = \mathbf{W}_{t-1} - \eta \nabla \mathcal{L}$ , where  $\mathbf{W}_t$  are the model weights at iteration  $t$ ,  $\mathbf{W}_{t-1}$  are the model weights in previous iteration,  $\eta$  is the learning rate, and  $\nabla \mathcal{L}$  are the gradients of the loss function. The learning rate is amount we should step in each iteration of SGD.
6. Repeat steps 1 through 5 until we converge to a minimum.

**Negative Sampling.** When we train the skip-gram model, we have to update weights for each word in the vocabulary. For large datasets, this can be a very expensive operation. Negative sampling reduces the computation required for training by only updating  $s$  words (which are *not* a context of the target word; hence these are *negative* examples). This means that we only update a small percentage of the model weights in each training iteration. Mikolov et. al. showed that for small datasets,  $5 \leq s \leq 20$  works best, whereas for large datasets,  $2 \leq s \leq 5$  suffice [27]. In our experiments, we use  $s = 5$ .

The negative samples are randomly sampled from the vocabulary with a probability distribution designed to favour more

frequent words. The probability of selecting a word  $w \in V$  as a negative sample is given by

$$P(w) = \frac{f(w)^\lambda}{\sum_{k=1}^{|V|} f(w_k)^\lambda}, \quad (1)$$

where  $f$  is the relative frequency function and  $\lambda$  is a hyper-parameter. In our experiments, we use  $\lambda = 0.75$ .

**Implementation.** We implement the model using the TensorFlow deep learning library. We chose TensorFlow since it is a versatile library for training and deploying neural networks [1]. It provides an interface for performing fast computation and linear algebra operations (such as matrix multiplication, which is core to training a neural network) on the GPU. It includes the `tf.keras` which simplifies the process of defining, training, and evaluating neural networks by providing implementations of common layers and activation functions such as 2D convolutional layers (`tf.keras.layers.Conv2D`), fully connected/linear layers (`tf.keras.layers.Dense`), the softmax activation function (`tf.keras.activations.softmax`), and more.

The `word2vec` module provides a straightforward implementation of the architecture discussed in this section as a `tf.keras` subclass (see the `Word2Vec` class). To efficiently load the corpus, we leverage the `tf.data` module. We use the `tf.data.Dataset` class to quickly load, tokenize, and batch our data (see the `make_dataset` function).

### 3.2 Sentiment Classification

In the last few years, transformer models have been shown to be widely effective on a variety of tasks in natural language processing. They have ushered in a new interest in applying transfer learning in natural language. In *Language Models are Few-shot Learners*, Brown et. al. showed that a massively scaled up stack of transformer-decoder layers consisting of 175 billion parameters (GPT-3) could achieve strong performance on many NLP tasks without any further model training.

**BERT & Transfer Learning.** The Bidirectional Encoder Representations from Transformer, or BERT for short is a similar general-purpose transformer model designed for transfer learning applications. The model is pre-trained to learn deep representations of text in an unsupervised manner. Then, the pre-trained model can be fine-tuned on a NLP tasks by simply adding a new output layer. Devlin et. al. showed that this approach can achieve state-of-the-art performance on various NLP tasks (such as question answering and language inference) [16].

Much of the state-of-the-art performance of these models comes from their main ingredient: the transformer [48]. Using an attention mechanism, transformers are able to model long-term dependencies in the data. At each training step, the attention mechanism allows the transformer to focus on a specific part of the input. In this way, attention is focus, which mimicks the natural way which we process information. The second ingredient in the success of these models is their size. By stacking many transformers on top one another, we are able

to increase the models capacity; however, this comes at a computational cost.

Transfer learning enables us to solve complex problems in natural language processing without a large dataset or the computational requirements of training a full-scale GPT or BERT model. For this reason, we use a pre-trained BERT model and then fine-tune it on the climate change sentiment dataset (see Section 2). Figure 8 illustrates the architecture of the classification model.

**Training.** The pre-trained BERT model checkpoints are provided by TensorFlow via TF Hub, a framework for performing transfer learning with TensorFlow [44]. We use the `tensorflow_hub` package to download and load the model, and then append a `tf.keras.layers.Dense` layer to the output of the BERT model to obtain our classification model [41, 42]. The model is trained to minimise the cross entropy loss (we use the `tf.keras.losses.SparseCategoricalCrossentropy` implementation), using the AdamW optimizer (from the `official.nlp.optimization` package) [43]. This is all contained within the `train_text_classifier.py` script.

**Interpreting Sentiment.** Our sentiment classification models predicts a probability distribution over all sentiment classes (i.e. positive, negative, and neutral). But, how can we produce a single sentiment value for an input? One approach is to take the class with the largest probability; however, what happens if two classes have similar probabilities? What if they differ by a very small factor? In this case, resolving the distribution to its maximum value might not accurately reflect the true class of the input. We address this problem by *sampling* from the probability distribution. If  $\hat{\mathbf{P}}$  is a probability vector representing the predicted probability distribution for  $m$  classes labelled  $\{1, 2, \dots, m\}$ , then for some  $1 \leq i \leq m$ , the probability of sampling class  $i$  is given by  $\hat{P}_i$ .

### 3.3 Embedding Projection

The ability to explore high-dimensional vector spaces such as word embeddings is crucial both in the development of models, and in the research of new techniques. Neural networks, embeddings, and other forms of highly-parameterised models are often referred to as ‘black boxes’ because studying the structure of these models is extremely difficult and sometimes impossible. At the same time, the ability to ‘peek’ into the internals is key in testing and debugging models. For example, we might want to verify that an embedding space is truly representing the desired features.

Compared to ‘conventional’ data, understanding word embeddings can be particularly challenging due to the high-dimensionality of the data. The sheer number of dimensions in the embedding space makes it impractical to work with the raw vectors. Similarly, conventional visualisation tools are seldom useful due to both a lack of support for large dimensional data, and a lack of interactivity which is key for navigating a dense vector space such as word embeddings. We follow an approach by Smilkov et. al. where we reduce the dimensionality of the embedding vectors to project it into a vector space which is



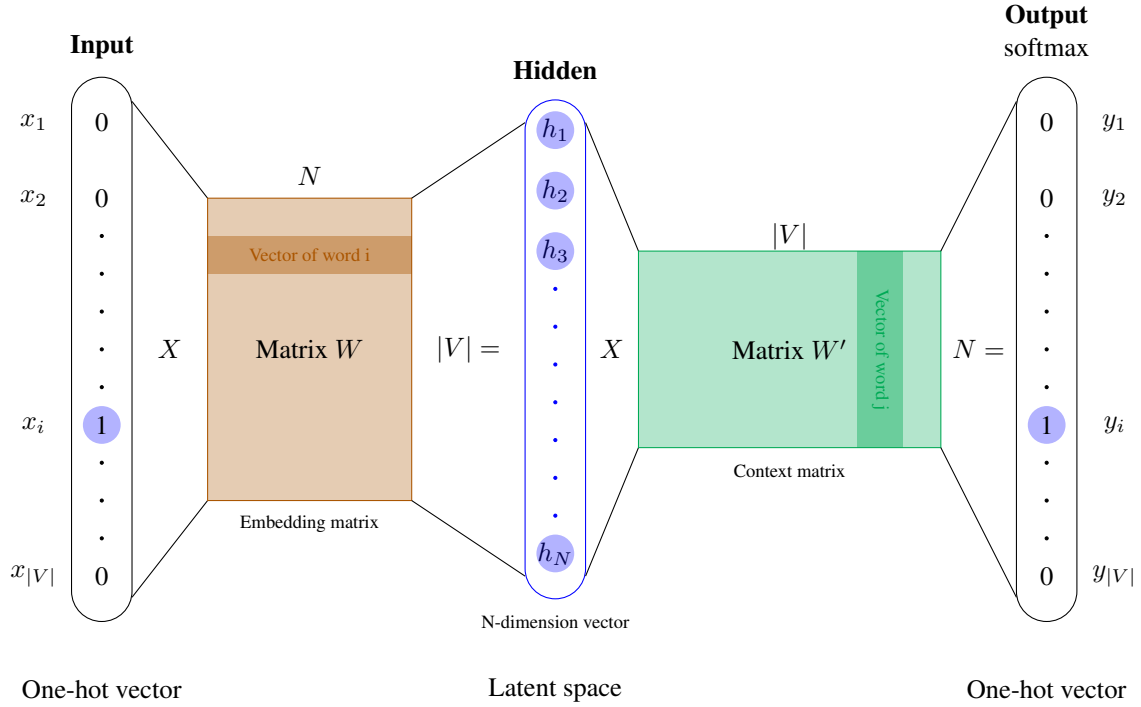


Figure 7: The architecture of the skip-gram model. Figure inspired by Weng [50].

human-friendly and easy to visualise [38]. In this section, we present an interactive visualisation and analysis tool for *projecting* high-dimensional vector spaces, such as word embeddings, into two- or three-dimensional space

**Overview.** The embedding projector is a Python application that can be ran from the command-line (see the `embedding_projector.py` module) built using the Plotly Dash framework [23]. Figure 9 shows the interface of the embedding projector. The application allows users to explore an arbitrary vector space given an embedding matrix; however, it is mostly tailored for use with word embeddings. The embedding projector allows the user to not only inspect the projected vector space, but also to search for words (in the analysis panel) and isolate the neighbouring points of a given selection (i.e. view a scatter plot only containing the neighbours).

**Principle Component Analysis.** To visualise the high-dimensional embedding vectors, we need to reduce it to a space that can be easily understood by a human: two- or three-dimensions. We use Principle Component Analysis (PCA), a common dimension reductionality approach to achieve this. For a given embedding matrix, we first compute the top ten principle components (these are the components with the most explained variance). Then, we select the top three of those components as our choice for the  $x$ ,  $y$ , and  $z$  dimensions of the projected space. While this means that the projected view is not an exact representation of the embedding geometry, it provides a very good approximation to the density of the space, which makes it ideal for analysing similarity and closeness of data points. The PCA panel (see Figure 9) allows users to configure which principle components correspond to which axis. We use

the `sklearn.decomposition.PCA` implementation which supports two algorithms for computing the principle components: taking the eigenvalues of the covariance matrix, or performing singular value decomposition (the algorithm for computing the PCA is chosen automatically) [31].

**Suffix Trees.** Searching the embedding space is crucial for adequate exploration of the data. For word embeddings, this means that we must be able to search for words in the vocabulary and find their corresponding embedding vector in the projected space. We say that a word  $w \in V$  is a search result if the query,  $q$ , is a substring of  $w$ .

A naive algorithm would iterate through the whole vocabulary, and check each word against the query. Of course, this is incredibly inefficient. For each query, we have to check against the *whole* vocabulary. Instead, we use suffix trees to efficiently perform substring searching on the vocabulary. This reduces the running time to  $\mathcal{O}(|q|)$ , at the cost of an initial startup of  $\mathcal{O}(|V|)$ , where  $|q|$  is the length of the query. For large vocabularies (such as the ones we train in this paper), this has a substantial performance difference.

We use a suffix tree implementation provided by the `suffix.trees` package [47]. Since suffix trees operate on a single string input, we convert our vocabulary into a space-separated string, and then feed that to the suffix tree. Queries on this suffix trees gives us a set of indices indicating where in the string the queried substring occurs. To resolve these matches into words, we exploit the structure of the string that the suffix tree was built on: the fact that it is space-separated. Thus, for each index, we search for the first space to the left and right of the match to determine the word. This has a runtime of  $\mathcal{O}(L)$ , where  $L$  is the length of the largest word in the vocabulary.



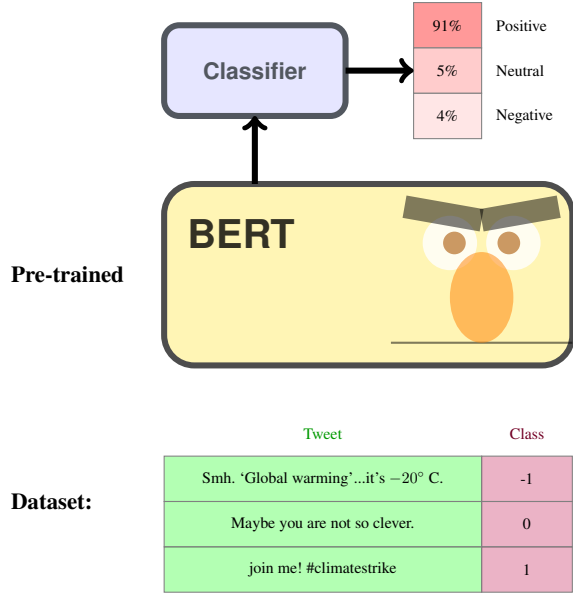


Figure 8: The architecture of the classification model. It uses a pre-trained BERT backbone with a fully-connected classification head with softmax activation. Figure inspired by Alammar [4].

Considering the fact that the length word in the English language is only 45 characters, and word frequency substantially decreases with length, this is not a bottleneck [39, 53].

**K-Nearest Neighbour Search.** We use a  $K$ -nearest neighbours model to efficiently find the most similar word embeddings. A naive ‘bruteforce’ algorithm to find the top  $k$  similar vectors would have to iterate through the whole vocabulary, compute the pairwise cosine similarity, and then sort the results, for a total running time of  $\Theta(|V| \times Nk)$ . In comparison, a  $k$ -nearest neighbours implemented as a KD-tree has a query runtime of  $\Theta(k \log |V|)$ , which is a vast improvement from the naive algorithm. This comes with a startup cost of  $\Theta(|V| \times N)$ , but it is negligible in the long-run. And the model implemented as a Ball tree has a similar run-time. We use the `sklearn.neighbors.KNeighborsClassifier` implementation which supports both KD-trees and Ball trees (automatically chosen based on the type of data) [31].

### 3.4 K-Hop Graphs: Visualising Word Relation

To perform an accurate analysis of tweet content, we must understand how words are used with one another, and in which contexts. Leveraging our word2vec models, we visualise the relationship between similar words as a graph, following the technique used by Schild et. al.

**Infinity-Hop Graph.** A  $\infty$ -hop graph is an undirected weighted graph whose nodes are words. It models the relationship between every word in the vocabulary. We say that for some distinct  $w_1, w_2 \in V$ , an edge with weight  $S(w_1, w_2)$ , between  $w_1$  and  $w_2$  exists if and only if  $S(w_1, w_2) \geq \alpha$ , where  $S$  is an arbitrary similarity metric between two words and  $\alpha$  is a

similarity threshold.

In this paper, we define  $S$  as the *cosine similarity* between the two word embedding vectors for words  $w_1$  and  $w_2$ . Geometrically, this similarity metric can be interpreted as the cosine of the angle between the two vectors.

**K-Hop Graphs.** We say that the  $k$ -hop graph for some word of interest,  $T \in V$ , is a subgraph of the  $\infty$ -hop graph, consisting of all the nodes and their respective edges in the  $\infty$ -hop graph that are at most  $k$  edges from  $T$ . Informally, this means that the  $k$ -hop graph is the  $\infty$ -hop graph limited to at most  $k$  ‘hops’ from a word of interest.

Formally, the  $k$ -hop graph,  $\mathcal{H}_k^{(T)}$ , for a vocabulary  $V$ , word of interest  $T \in V$ , similarity function  $S : V^2 \rightarrow \mathbb{R}$ , and similarity threshold  $\alpha \in \mathbb{R}$ , is a subgraph of the  $\infty$ -hop graph defined by the ordered pair

$$\mathcal{H}_k^{(T)} = (N_k^{(T)}, E_k^{(T)}),$$

where  $N_k^{(T)} \subseteq N_\infty$  is the node or vertex set, and  $E_k^{(T)} \subseteq E_\infty$  is the edge set, where  $N_\infty$  and  $E_\infty$  are the node and edge sets for the  $\infty$ -hop graph respectively.

For all  $w \in N_\infty$ , the word  $w$  is a node of the  $k$ -hop graph if and only if there exists a path with at most  $k$  edges from  $w$  to  $T$  in the  $\infty$ -hop graph. The edges of  $\mathcal{H}_k^{(T)}$  are simply those which are leftover after filtering out nodes which are not reachable from  $T$  in at most  $k$  edges. So,

$$E_k^{(T)} = \{(w_1, w_2) \in E_\infty \mid w_1, w_2 \in N_k^{(T)}\}.$$

**Visualising the Graph.** We employ several algorithms for visualising the  $k$ -hop graphs. To build, process, and manipulate the graphs, we use the NetworkX Python package—a popular library for studying graphs and networks [15]. After computing the node and edge set of the  $k$ -hop graph, we use a force-directed layout algorithm called ForceAtlas2 (using the `fa2` package) to spatialize the network [13, 24]. The algorithm takes into account edge weights to determine an optimal graph layout. Then, we perform Louvain community detection (using the `python-louvain` package) to extract communities from the network in an unsupervised manner [8, 10]. We assign a colour to each community, and colour each node with that colour. Edges colours are determined by the colour of the source node.

**Implementation Details.** Building the infinity-hop graph for the projection weights matrix,  $\mathbf{W}^{|V| \times N}$  (where  $|V|$  is the vocabulary size and  $N$  is the embedding dimensionality) requires the computation of a pairwise cosine similarity matrix,  $\mathbf{M}$ . For large vocabularies, computing  $\mathbf{M}$  is a particularly expensive operation, since it requires  $\Theta(|V|^2)$  space. For a matrix with 32-bit floating-point values (`float32`), this can easily exceed 40 GB ( $|V| \geq 100,000$ ).

However, notice that we don’t actually need to store anything: building the infinity-hop graph simply requires that we keep the pairs whose similarity is above a threshold. So, we reduce the memory footprint by *batching* the computations. For some batch size  $B \in \mathbb{N}$ , we iteratively construct the infinity-hop graph by computing the pairwise cosine similarity matrix

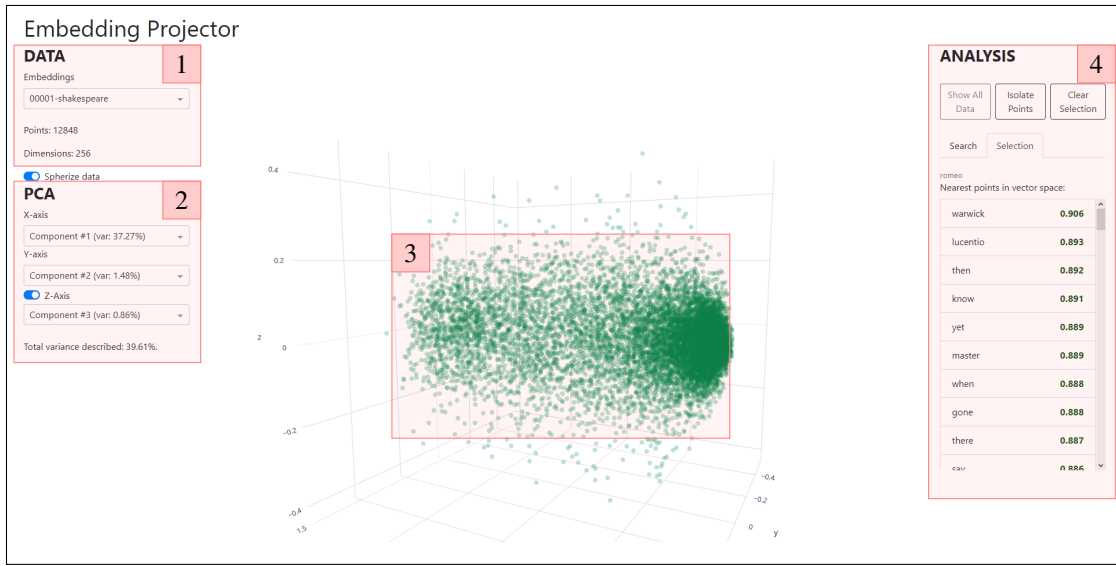


Figure 9: The interface of the embedding projector: (1) the data panel, (2) the PCA panel, where users can choose any combination of two or three principle components, (3) the view of the projected data, and (4) the analysis panel, where users can search for words and explore their neighbours in the embedding space.

between  $\mathbf{W}$  and  $B$  rows of  $\mathbf{W}$  at a time. More specifically, for some row offset  $i \leq \left\lfloor \frac{|V|}{B} \right\rfloor + 1$ , we compute the pairwise cosine similarity matrix,  $\mathbf{M}'$  as the normalized dot product of a smaller matrix and  $\mathbf{W}$ , given by

$$\mathbf{M}' = \frac{\langle \mathbf{C}^{(i)}, \mathbf{W} \rangle}{\|\mathbf{C}^{(i)}\|_2 \|\mathbf{W}\|_2},$$

where  $\mathbf{C}^{(i)}$  is a matrix containing all rows of  $\mathbf{W}$  with indices between  $iB$  and  $(i+1)B$ . Then, we apply filtering on  $\mathbf{M}'$  to build the infinity-hop graph. Converting rows from ‘local’ indices to their word embedding index just requires adding the row offset. For example, the first row of  $\mathbf{M}'$  corresponds to the  $B$ -th row of  $\mathbf{W}$ , and in general the  $j$ -th row of  $\mathbf{M}'$  is the  $j + Bi$ -th row of  $\mathbf{W}$ , where  $i$  is the row offset.

This approach yields a space complexity of  $\Theta(|V| \times B)$ . For  $B = 256$  and  $|V| = 100,000$ , this means that we only need roughly 102 MB of available RAM—a significant reduction from the 40 GB requirement of the previous algorithm.

### 3.5 Instructions for Reproducibility

We provide the source code for running all experiments, along with the training and testing data, and pre-trained model weights.<sup>2</sup>

**Note About Requirements.** Some of the required packages include native code. Unfortunately, these libraries do not come with pre-compiled binaries for Python 3.8 so a C/C++ compiler is required to build them from source when installing requirements. On Windows, download [MSVC 1.14+](#). On Mac or Linux, gcc or g++ will work.

<sup>2</sup><https://github.com/GalacticGlum/csc110-course-project>

**Dataset.** Click [here](#) to download the data used to train and evaluate the models. Note that this is a fairly large file (roughly 4 GB compressed), so downloading may take a while depending on internet connection. Extract the archive in a folder called data (under the root project directory).

**Model Weights.** Click [here](#) to download the **full** model checkpoints and weights. Note that this is a fairly large file (roughly 20 GB compressed), so downloading may take a while depending on the internet connection.

For this reason, we also provide a subset of the trained model weights (roughly 1 GB compressed), which contain the word embeddings for tweets posted in April, May, and June, 2020 along with the trained sentiment classification model. We also provide light-weight word embeddings (roughly 50 MB) for a corpus consisting of Shakespeare’s works which were used for testing the models and visualisation tools.

Click [here](#) to download a **subset** of the model checkpoints and weights. In the interest of space, we omit all checkpoints except for the latest one. Furthermore, for the word2vec models, since the output layer is not used outside of training, we omit the checkpoints entirely. Instead, we only provide the projection weights (proj\_weights.npy), model vocabulary (vocab.txt), and tokenizer state (tokenizer.json).

After the download is complete, extract the archive in the root project directory. The zip file already contains an output sub directory, so there is no need to create one.

**Running the Experiments & Visualisations.** See the provided main.py for starter code to run the embedding projector, and/or generate a  $k$ -hop graph.

Major Events		
#	Day	Event
1	June 8, 2018	The 44th G7 summit is held in Canada. Donald Trump gets into multiple disputes with other world leaders: insults Japanese Prime Minister Shinzo Abe, and speaks harshly about Canadian Prime Minister Justin Trudeau [51].
2	September 14, 2018	Global Climate Action Summit is held [30].
3	August 19, 2019	NASA Confirms the uptick in Amazon rainforest fire activity
4	March 15, 2019	A series of global school strikes. The first organised by Greta Thunberg. Over 1.4M people participate. [54].
5	September 20, 2019	Global week of climate action. Over 4M people participate. [54].
6	November 4, 2019	The Trump administration gives a formal notice of intention to withdraw from the Paris Agreement [55].
7	June 12, 2020	The 46th G7 summit is cancelled due to COVID-19 [52].

Table 2: Major climate events. Figures 10-13 are annotated with these events.

## 4 Results

### 4.1 Temporal Analysis

We start by analysing how basic features of tweets have changed over time. More specifically, we study how the frequency and proportion of certain tweets have changed from a temporal perspective, while also putting it into context with key events that have occurred in the last two years. See Table 2 for a summary of the real-world events which we consider in this analysis.

**Evolution of Climate Change Discussion.** First, we look “climate change” and its (outdated) sibling, “global warming.” We use these two terms to gauge how general discussion about climate change has evolved over time (under the assumption that they are proxies for climate change discussion). Figure 10 shows the monthly mentions of the terms “climate change” and “global warming”. First, we observe that both the frequency and proportion for “global warming” is markedly lower, and its curve is flatter, compared to the frequency and proportion distributions for “climate change”; however, the curves for “global warming” and “climate change” follow roughly the same shape, which indeed suggests that they are synonymous terms used in place with one another. This makes sense since “global warming” is a term which has fallen out of fashion in the last decade [29]. Seeing as a majority of Twitter users are of the younger demographic, it is consistent that they would use “climate change” as opposed to its outdated counterpart [29].

Next, we notice that the mentions for both words sharply increases after June 8, 2019, which corresponds to the first day of the 44th G7 summit [51]. This suggests that Twitter users may have been discussing the summit and as it relates to climate change. However, we do not see a similar jump in mentions on September 14, 2018—when the Global Climate Action Summit was held. This could be due to the fact that the summit did not attract as much attention compared to the G7 summit. Indeed, it plausible that controversies during the G7 summit attracted the attention of Twitter users, which therefore also brought more

attention to the climate change issue (relating to U.S. President Donald Trump making off-hand remarks to other world leaders) [51]. In comparison, there were no such controversies during the Global Climate Action Summit [30].

Similarly, we observe the most sizable jump after the global school strikes held on March 15, 2019 [54]. This makes sense since Greta Thunberg used Twitter and other social media very heavily to market the movement. This is reflected in the fact that the number of mentions (both in frequency and proportion) for “climate strike”, “climate justice”, “climate activists”, and “climate action” experienced a spike around September 20, 2019 (see Figure 11). Likewise, in Figure 13, we see that there is a very large spike in usage of “greta thunberg” on Twitter after the around March, 2019, which was when the first global school strikes were organised [54].

We also see an initial spike in the usage of “climate crisis” near August, which might correspond the uptick in Amazon rainforest fire activity which began to receive international attention in late-July to early-August [25].

**Climate Change Skepticism.** To get a grasp on how climate change skepticism might have evolved on Twitter, we look at the frequency and proportion distribution for the following terms: “#climatechangehoax”, “skeptic”, “deniers”, and “denier”, and “#climatechangeisreal”, which were selected based on their sentiment and proximity in the embedding vector space. From Figure 12, we see that the usage of “#climatechangehoax” and “#climatechangeisreal” is very sparse and almost non-existent. On the other hand, we observe a spike in the usage of “deniers”, “denier”, and “skeptic” all around the third and fourth event markets, which correspond to the uptick in Amazon rainforest fire activity and the global school action strikes organised by Greta Thunberg. This suggests that there was backlash from the skeptic climate change community following these events. And this is consistent with our assumptions: increased activity by climate change skeptics after such events means that the community was attempting to dismiss these events—a common pattern among skeptic communities.

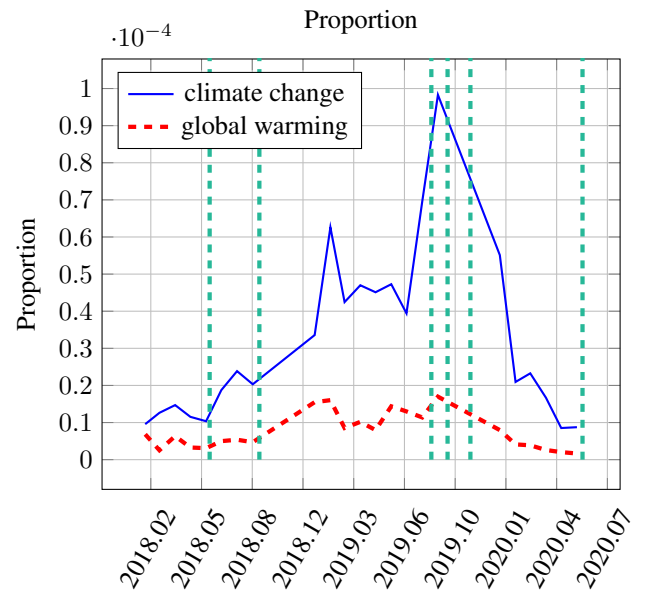
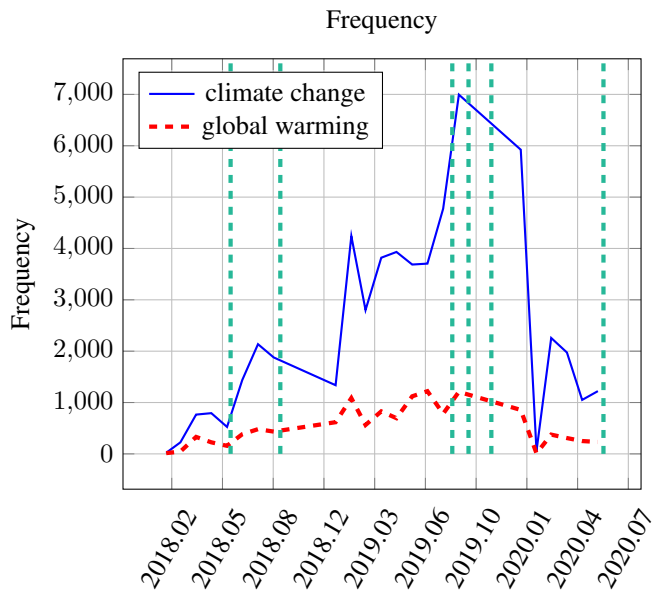


Figure 10: Monthly mentions of the terms “climate change” and “global warming” on Twitter.

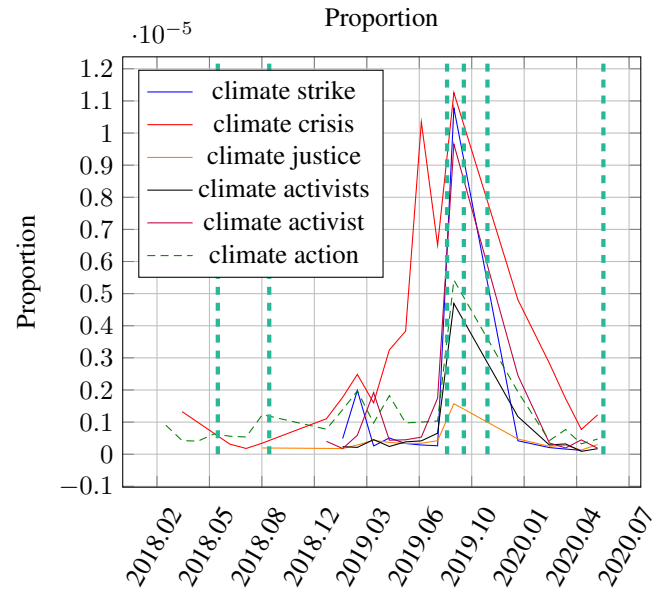
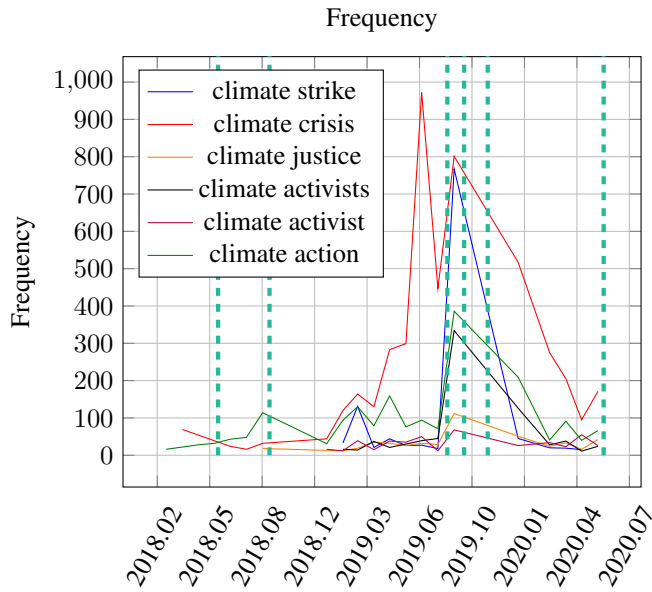


Figure 11: Monthly mentions of the terms “climate strike”, “climate justice”, and “climate activists”, “climate activist”, and “climate action” on Twitter.

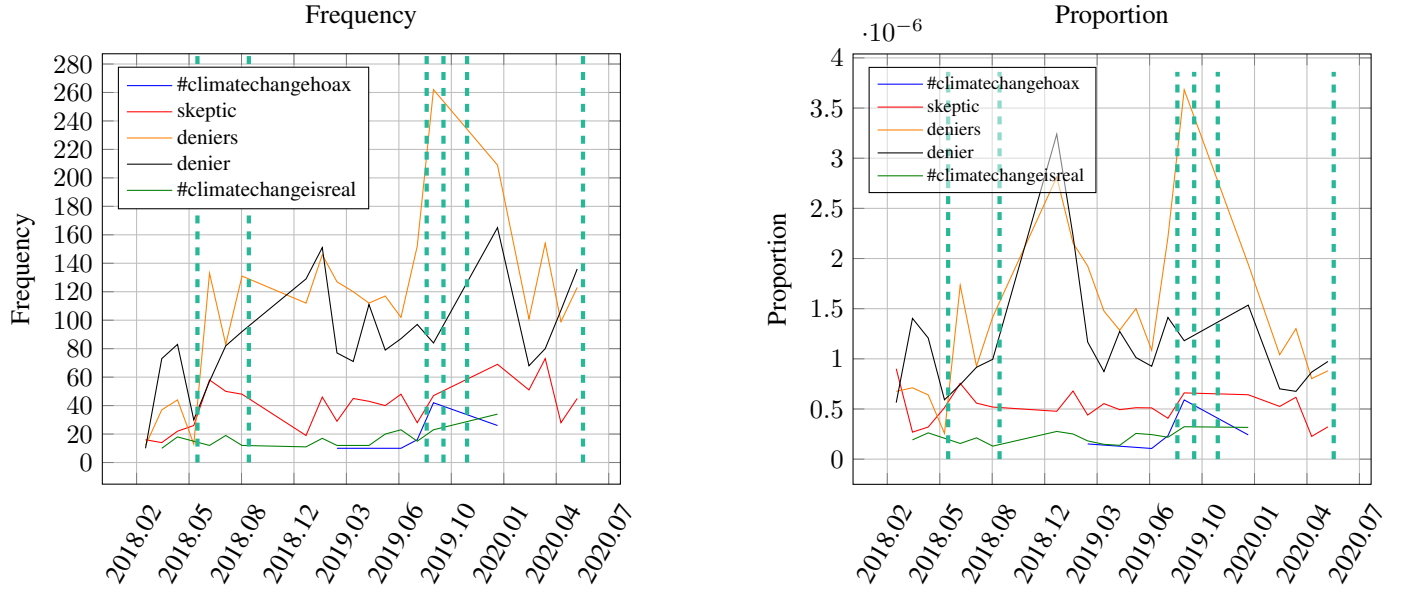


Figure 12: Mentions of the terms “#climatechangehoax”, “skeptic”, “deniers”, “denier”, and “#climatechangeisreal” on Twitter.

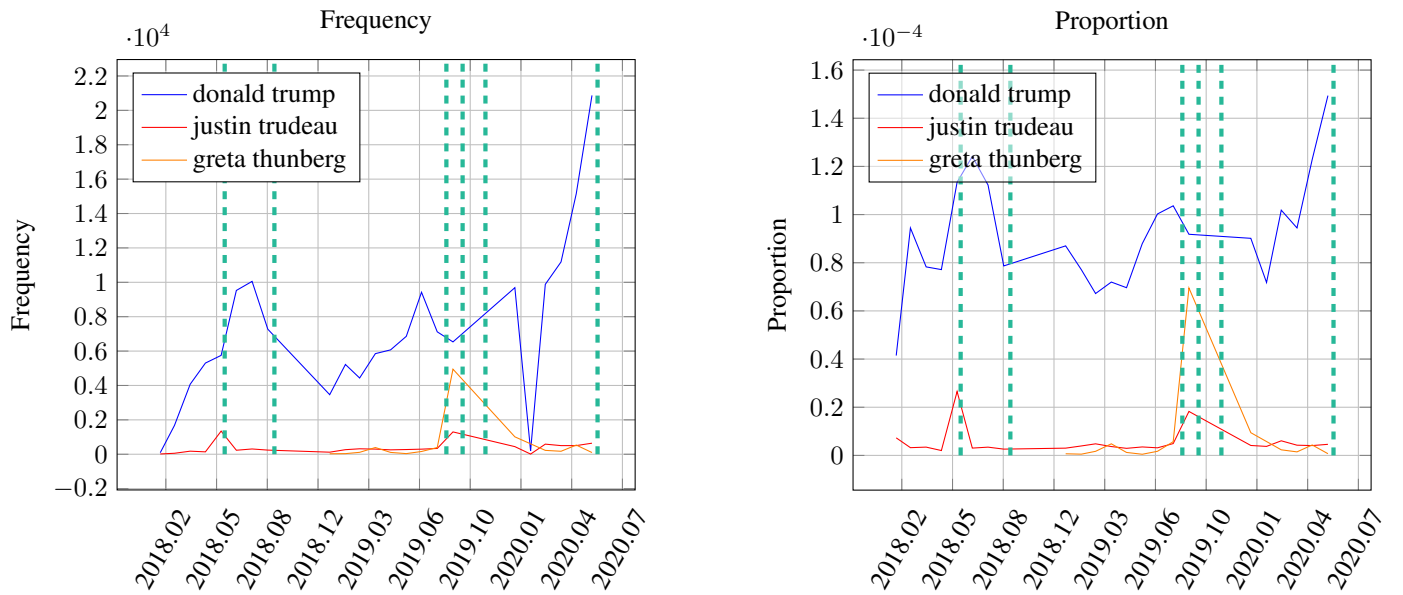


Figure 13: Mentions of the people “donald trump”, “justin trudeau”, and “greta thunberg” on Twitter.

## 4.2 Content Analysis

We train a word2vec model on each monthly archive of the Twitter dataset (see Section 2). Using this ensemble of word2vec models, we can analyse the content of tweets.

**Contextual Evolution.** First we look at the evolution of word context over time. Table 3 depicts the top 20 most similar words to “climate\_change” and “global\_warming” for the word2vec models trained on the first and last month of data available. The first observation we make is that words are *more* similar to “climate\_change” and “global\_warming” in the last word2vec model compared to the first. This indicates that the neighbourhood around these terms is *denser* in the embedding space for June, 2019. Next, we notice that compared to February 2018, the top 20 most similar words in June, 2020 have taken on a more dire sentiment. While for the first month, the most similar word to “climate\_change” was “opposition”, in the last month, the most similar word is “climate\_crisis”. This change in word usage over time indicates a general shift in climate change sentiment on Twitter—it shows how the seriousness of the situation has taken root, and it suggests that in climate change related discussions, people are using more negative words with more serious undertones.

Furthermore, we see that climate change and global warming are near COVID-19 related terms—“covid\_19”, “pandemic”, and other similar variants show up very near the top of the list. Indeed, this does suggest a hidden relationship between the pandemic and climate change. This could be due to the fact that they are both—in their own right—global emergencies, and such they appear in similar contexts.

Next, we study how the cosine similarity of certain word pairs have changed over time. Figures 14 and 15 depict the cosine similarity of “climate\_change” to various terms related to climate change skepticism/denial, climate crisis, and other serious words. First, we observe that the cosine similarity of “climate\_change” and “denier” is fairly volatile, but has been decreasing since late 2019. We observe a similar trend for the cosine similarity of “climate\_change” and “hoax”. However, for the rest of the terms in Figure 14, we notice a somewhat peculiar trend. To start, we can see that the similarity curves for “fake“, “donald.trump“, “maga“, and “liberal” all follow similar shapes. This could be an indication that there is an even deeper relationship between all these terms—something external is causing the similarity of these terms to change all at once. One possible explanation for this goes back to the fact that climate change discussion tends to be highly politicized. So, it is not too surprising that we observe a similar curves for the political terms (such as “donald.trump“, “maga“, etc...). Secondly, we observe that all the curves seems to spike at around the same time: February/March, 2020. This corresponds with the impetus of the COVID-19 pandemic, likely indicating that discussions about the pandemic led to discussions about climate change. This supports our findings in the Table 3, where pandemic-related terms were in the top 20 most similar list of words to “climate\_change” and “global\_warming”.

Shifting our focus onto Figure 15, we see that the similarity for words with more dire sentiments has been growing over

time. Specifically, we see a steady similarity for “poverty”, “homelessness”, and “inequality”. This likely indicates how Twitter users feel about what is *causing* climate change, and shows a fairly strong negative sentiment. Likewise, we observe that the similarity for “mitigation”, “preventable”, and “unavoidable” are also very high. Once again, the steady trend in the similarity of these terms suggests that on the whole, Twitter users feel that climate change mitigation and prevention is important. Looking back to our temporal analysis, this supports the fact that there was an increase in mentions about climate change activism (see Figure 11).

**Word Relation.** Figures 16 and 17 illustrate the 2-hop graph for the first and last word2vec models. In the first figure, we see that the a majority of the words are negative, and that there are two main communities: one with words such as “struggling”, “stressing”, and “insecure”, and the other with words such as “outrageous”, “endure”, and “unhappy”. This shows a worrisome trend of negativity surrounding climate change. On the other hand, the second figure depicts far more topical words. We see that the largest nodes, and thus those with the most connections, are words such as “#climatechange”, “socio\_economic”, and “mitigating”, which suggests a much more productive stance towards climate change. As a whole, from this pair of 2-hop graphs, we can see a shift in attitudes about climate change on Twitter—from a purely negative stance, to one that is about resolution, though with an increased urgency. Interestingly, the two communities in the second graph show a very clean split between terms: the community coloured orange mostly contains terms related to technology, whereas the blue community contains terms related to emergency, crisis, and inequity. We can see the interaction between these two related concepts in Figure 17.

## 4.3 Sentiment Analysis

Using the sentiment classification model, we analyse the climate change sentiment of tweets. For each month, we take a 10% random sample of all the tweets in the dataset, and then compute the climate change sentiment of each tweet (according to methodology presented in Section 3). Figure 18 shows the average climate change sentiment per month, plotted on a  $-1$  (negative) to  $1$  (positive) scale. We see that the sentiment mostly hovers 0, which suggests that most tweets are neutral. After all, this makes sense since Twitter is a general discussion platform, and the reality is that discussions about climate change are in the minority of all discussions. However, around February/March, 2020, we see an a fairly substantial increase in the climate change sentiment. This suggests not only an increase in climate change related activity during that period of time, but also *positive* discussion. Notice that this is the same time period for which we observed the spike in the cosine similarity graphs (Figure 14). On the other hand, we also see a dip in the climate change sentiment around June, 2019. Recall from Table 2 that this was around the time when the 46th G7 summit was *supposed* to be held. Thus, this could suggest a response to the summit, where there was an increase in the activity of climate change skeptics.

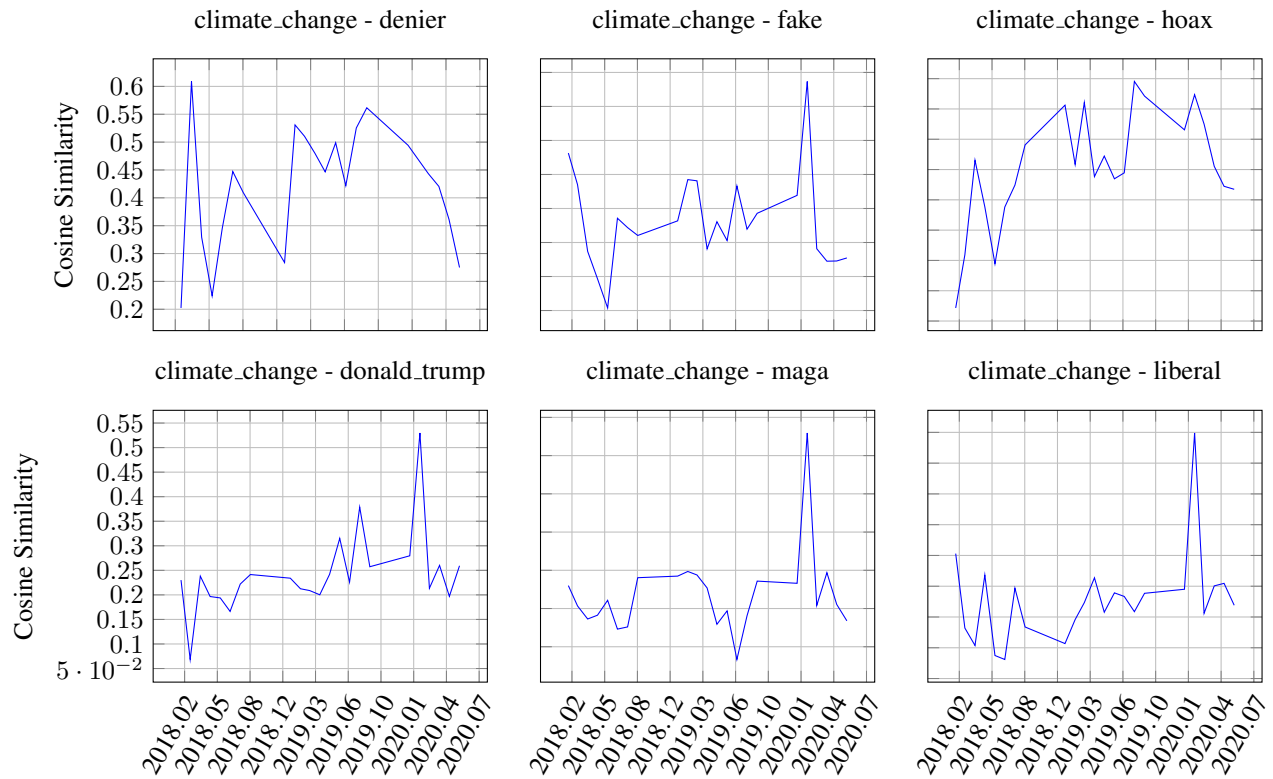


Figure 14: Cosine similarity of “climate\_change” and various terms related to climate change skepticism/denial and politics.

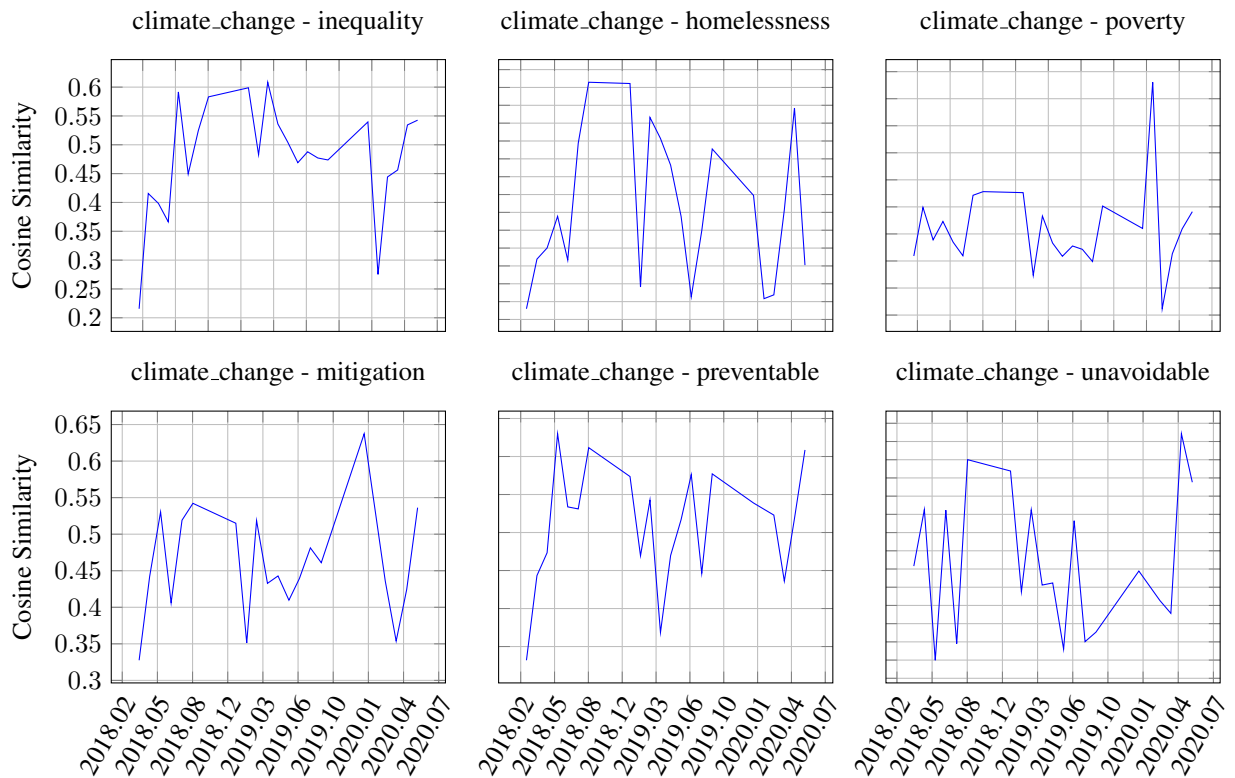
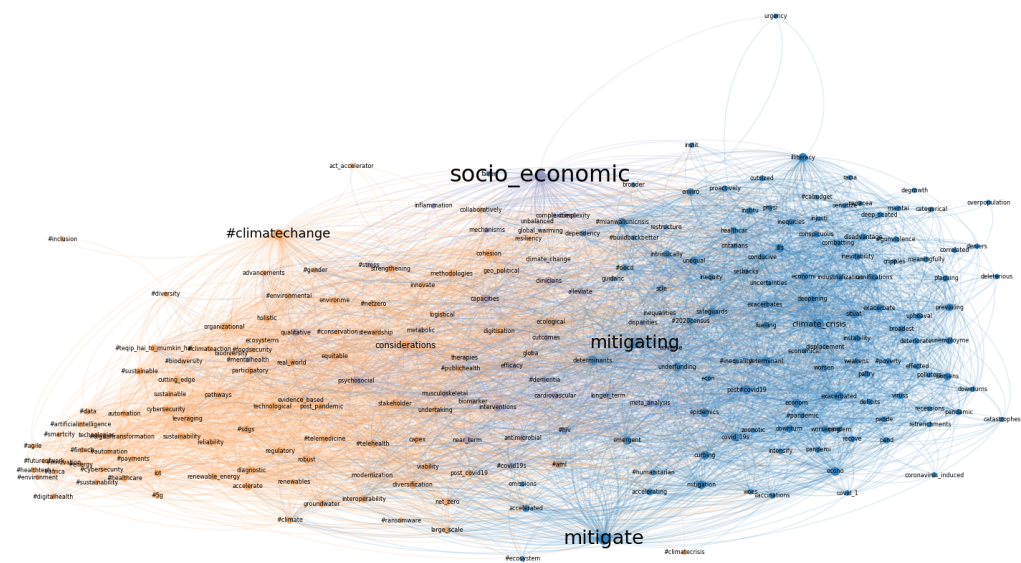
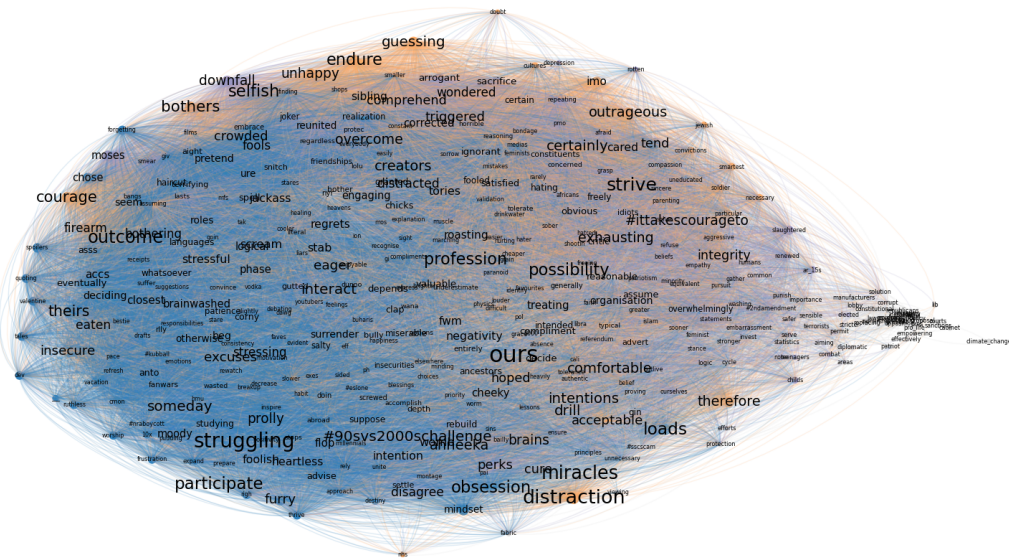


Figure 15: Cosine similarity and “climate\_change” to various serious terms.





First Word2Vec Model (month of February, 2018)				Last Word2Vec Model (month of June, 2020)			
Word (climate_change)	Similarity	Word (global_warming)	Similarity	Word (climate_change)	Similarity	Word (global_warming)	Similarity
opposition	0.526	moronic	0.543	climate_crisis	0.651	climate_change	0.605
ours	0.516	guessing	0.537	#climatechange	0.639	unbalanced	0.601
greater	0.495	unfortunate	0.53	global_warming	0.605	#poverty	0.596
4 ourselves	0.487	planted	0.533	pollution	0.595	epidemics	0.591
integrity	0.486	miracles	0.509	covid_19s	0.574	adverse	0.589
viewing	0.481	plain	0.506	epidemics	0.572	viruss	0.587
advert	0.478	smartest	0.504	#inequality	0.558	deepened	0.586
muc	0.475	nyr	0.501	mitigate	0.557	persistent	0.584
movements	0.475	believes	0.500	near_term	0.555	mutation	0.584
necessary	0.471	johns	0.497	ecological	0.550	illiteracy	0.583
proving	0.471	lt2018	0.496	#poverty	0.546	tuberculosis	0.581
intended	0.470	everton	0.495	inequality	0.543	catastrophic	0.580
imo	0.467	critic	0.493	mitigation	0.536	preventable	0.580
humanity	0.466	hired	0.490	disparities	0.536	situation	0.576
states	0.466	#megafree	0.490	emissions	0.535	pandemics	0.576
		checkupcamp					
logical	0.465	appears	0.484	obesity	0.532	susceptibility	0.575
heartless	0.465	begins	0.480	meta_analysis	0.532	cv19	0.575
farmers	0.465	competitor	0.479	#pregnancy	0.530	worsens	0.575
embrace	0.465	trains	0.477	cardiovascular	0.525	mutations	0.575
buhari	0.463	mat	0.477	mutation	0.523	unavoidable	0.574

Table 3: The top 20 most similar words to “climate\_change” and “global\_warming” for the first and last word2vec models.

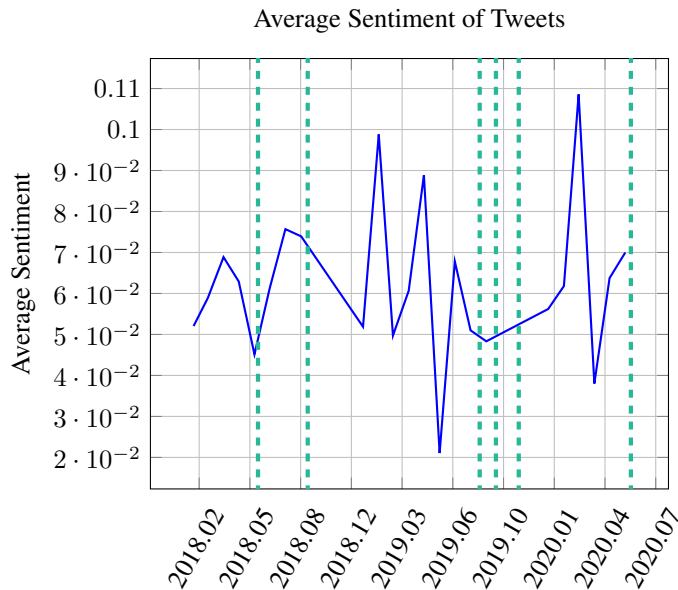


Figure 18: The average climate change sentiment of a 10% random sample per month, plotted on a  $-1$  to  $1$  scale. The score of a tweet is assigned as follows:  $1$  if positive,  $-1$  if negative, and  $0$  if neutral.

## 5 Discussion

Based on the results of the temporal, content, and sentiment analysis, we see how the climate change sentiment of Twitter users has been shifting to one that is more serious, polarized; it echos the dire situation, and the current state of the climate emergency. As shown in the Section 4, there is a high emphasis on climate change on Twitter. We also see how discussions about climate change on Twitter have become more ‘severe’, and that tweets are more likely to be of a positive sentiment (i.e. supporting climate change); that is, to be associated with the ‘great’ or ‘good’ aspects of climate issue—as opposed skepticism and climate change denial.

Further research is desired to investigate how misinformation on Twitter is targeted to affect the sentiment of users. This remains inconclusive

## References

- [1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Software available from tensorflow.org. 2015. URL: <http://tensorflow.org/>.
- [2] United States Environmental Protection Agency. *Inventory of U.S. Greenhouse Gas Emissions and Sinks*. <https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks>. 1990-2018.
- [3] Taufiq Ahmad, Aima Alvi, and Muhammad Ittefaq. “The Use of Social Media on Political Participation Among University Students: An Analysis of Survey Results From Rural Pakistan”. In: *SAGE Open* 9.3 (July 2019), p. 215824401986448. DOI: 10.1177/2158244019864484. URL: <https://doi.org/10.1177/2158244019864484>.
- [4] J. Alammr. *The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning)*. <https://jalammr.github.io/illustrated-bert/>.
- [5] Ashley A. Anderson. “Effects of Social Media Use on Climate Change Opinion, Knowledge, and Behavior”. In: (Mar. 2017). DOI: 10.1093/acrefore/9780190228620.013.369. URL: <https://doi.org/10.1093/acrefore/9780190228620.013.369>.
- [6] Monica Anderson, Skye Toor, Lee Rainie, and Aaron Smith. *Public attitudes toward political engagement on social media*. <https://www.pewresearch.org/internet/2018/07/11/public-attitudes-toward-political-engagement-on-social-media/>. 2018.
- [7] The Internet Archive. *Archive Team, Twitter Stream monthly archives*. <https://archive.org/search.php?query=collection%3Atwitterstream&sort=-publicdate>.
- [8] Thomas Aynaud. *Louvain Community Detection*. <https://github.com/taynaud/python-louvain>. 2020.
- [9] Matthew T. Ballew, Anthony Leiserowitz, Connie Roser-Renouf, Seth A. Rosenthal, John E. Kotcher, Jennifer R. Marlon, Erik Lyon, Matthew H. Goldberg, and Edward W. Maibach. “Climate Change in the American Mind: Data, Tools, and Trends”. In: *Environment: Science and Policy for Sustainable Development* 61.3 (Apr. 2019), pp. 4–18. DOI: 10.1080/00139157.2019.1589300. URL: <https://doi.org/10.1080/00139157.2019.1589300>.
- [10] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. “Fast unfolding of communities in large networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2008.10 (2008), P10008. ISSN: 1742-5468. DOI: 10.1088/1742-5468/2008/10/p10008. URL: <http://dx.doi.org/10.1088/1742-5468/2008/10/P10008>.
- [11] M. Boykoff, J. Katzung, and A. Nacu-Schmidt. *A Review of Media Coverage of Climate Change and Global Warming in 2019*. [https://sciencepolicy.colorado.edu/icecaps/research/media coverage/summaries/special issue 2019.html](https://sciencepolicy.colorado.edu/icecaps/research/media%20coverage/summaries/special%20issue%2019.html). 2020.
- [12] Pew Research Center. *Global Attitudes & Trends*. Data retrieved from <https://www.pewresearch.org/global/datasets/>. 2006–2018.
- [13] Bhargav Chippada. *ForceAtlas2 for Python*. <https://github.com/bhargavchippada/forceatlas2>. 2020.
- [14] John Cook. “Understanding and Countering Misinformation About Climate Change”. In: *Advances in Media, Entertainment, and the Arts*. IGI Global, 2019, pp. 281–306. DOI: 10.4018/978-1-5225-8535-0.ch016. URL: <https://doi.org/10.4018/978-1-5225-8535-0.ch016>.
- [15] NetworkX developers. *NetworkX*. <https://networkx.org/>. 2020.
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2019. arXiv: 1810.04805 [cs.CL].
- [17] J. Eilperin, D. Brady, and J. Muyskens. *Trump rolled back more than 125 environmental safeguards. Here’s how*. <https://www.washingtonpost.com/graphics/2020/climate-environment/trump-climate-environment-protections/>. 2020.
- [18] Moira Fagan and Christine Huang. *A look at how people around the world view climate change*. <https://www.pewresearch.org/fact-tank/2019/04/18/a-look-at-how-people-around-the-world-view-climate-change/>. 2019.

- [19] Cary Funk and Meg Hefferson. *U.S. Public Views on Climate and Energy*. <https://www.pewresearch.org/science/2019/11/25/u-s-public-views-on-climate-and-energy/>. 2019.
- [20] Yoav Goldberg and Omer Levy. *word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method*. 2014. arXiv: 1402.3722 [cs.CL].
- [21] J. Gottfried and E. Shearer. *News Use Across Social Media Platforms 2016*. <https://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/>. Pew Research Center, 2016.
- [22] Jeffrey Gottfried and Elisa Shearer. *News Use Across Social Media Platforms*. <https://www.journalism.org/2016/05/26/news-use-across-social-media-platforms-2016/>. 2016.
- [23] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: <https://plot.ly>.
- [24] Mathieu Jacomy, Tommaso Venturini, Sebastien Heymann, and Mathieu Bastian. "ForceAtlas2, a Continuous Graph Layout Algorithm for Handy Network Visualization Designed for the Gephi Software". In: *PLoS ONE* 9.6 (June 2014). Ed. by Mark R. Muldoon, e98679. DOI: 10.1371/journal.pone.0098679. URL: <https://doi.org/10.1371/journal.pone.0098679>.
- [25] NASA's Jet Propulsion Laboratory. *Scientific Consensus: Earth's Climate is Warming*. <https://climate.nasa.gov/scientific-consensus/>. 2020.
- [26] Anthony Leiserowitz. "Climate Change in the American Mind: National Survey Data on Public Opinion (2008-2018)". In: (2020). DOI: 10.17605/OSF.IO/JW79P. URL: <https://osf.io/jw79p/>.
- [27] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013. arXiv: 1301.3781 [cs.CL].
- [28] Brett Molina. *Twitter overcounted active users since 2014, shares surge on profit hopes*. <https://www.usatoday.com/story/tech/news/2017/10/26/twitter-overcounted-active-users-since-2014-shares-surge/801968001/>.
- [29] NASA. *What's in a Name? Global Warming vs. Climate Change*. <https://www.nasa.gov/topics/earth/features/climate-by-any-other-name.html>.
- [30] The United Nations. *Global Climate Action Summit Brings Surge of New Commitments*. <https://www.un.org/sustainabledevelopment/blog/2018/09/global-climate-action-summit-brings-surge-of-new-commitments/>.
- [31] 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. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.
- [32] Francesco Pierri and Stefano Ceri. *False News On Social Media: A Data-Driven Survey*. 2020. arXiv: 1902.07539 [cs.SI].
- [33] E. Qian. *Twitter Climate Change Sentiment Dataset*. <https://www.kaggle.com/edqian/twitter-climate-change-sentiment-dataset>. 2019.
- [34] H. Ritchie and M. Roser. "CO<sub>2</sub> and Greenhouse Gas Emissions". In: *Our World in Data* (2017). <https://ourworldindata.org/co2-and-other-greenhouse-gas-emissions>.
- [35] A. Samantray and P. Pin. "Credibility of climate change denial in social media". In: *Palgrave Communications* 5.1 (Oct. 2019). DOI: 10.1057/s41599-019-0344-4. URL: <https://doi.org/10.1057/s41599-019-0344-4>.
- [36] Leonard Schild, Chen Ling, Jeremy Blackburn, Gianluca Stringhini, Yang Zhang, and Savvas Zannettou. "Go eat a bat, Chang!": An Early Look on the Emergence of Sinophobic Behavior on Web Communities in the Face of COVID-19. 2020. arXiv: 2004.04046 [cs.SI].
- [37] K. Shu, A. Sliva, S. Wang, J. Tang, and H. Liu. "Fake News Detection on Social Media". In: *ACM SIGKDD Explorations Newsletter* 19.1 (Sept. 2017), pp. 22–36. DOI: 10.1145/3137597.3137600. URL: <https://doi.org/10.1145/3137597.3137600>.
- [38] Daniel Smilkov, Nikhil Thorat, Charles Nicholson, Emily Reif, Fernanda B. Viégas, and Martin Wattenberg. *Embedding Projector: Interactive Visualization and Interpretation of Embeddings*. 2016. arXiv: 1611.05469 [stat.ML].
- [39] Reginald D. Smith. *Distinct word length frequencies: distributions and symbol entropies*. 2012. arXiv: 1207.2334 [cs.CL].
- [40] United States Geological Survey. *How much carbon dioxide does the United States and the World emit each year from energy sources?* <https://www.usgs.gov/faqs/how-much-carbon-dioxide-does-united-states-and-world-emit-each-year-energy-sources?qt-news-science-products>. 2017.
- [41] TensorFlow. *Module: tf.Keras*. [https://www.tensorflow.org/api\\_docs/python/tf/keras](https://www.tensorflow.org/api_docs/python/tf/keras).
- [42] TensorFlow. *TensorFlow Hub*. <https://github.com/tensorflow/hub>.
- [43] TensorFlow. *TensorFlow Model Garden*. <https://github.com/tensorflow/models>.

- [44] *TensorFlow Hub*. <https://www.tensorflow.org/hub>.
- [45] Twitter. *Counting characters when composing Tweets*. <https://developer.twitter.com/en/docs/counting-characters>.
- [46] Twitter. *Sampled stream v1 (retired)*. <https://developer.twitter.com/en/docs/labs/sampled-stream/overview>.
- [47] Peter Us. *suffix\_trees*. <https://github.com/ptrus/suffix-trees>.
- [48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].
- [49] Soroush Vosoughi, Deb Roy, and Sinan Aral. “The spread of true and false news online”. In: *Science* 359.6380 (Mar. 2018), pp. 1146–1151. DOI: 10.1126/science.aap9559. URL: <https://doi.org/10.1126/science.aap9559>.
- [50] Lillian Weng. *Learning Word Embeddings*. <https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html#context-based-skip-gram-model>.
- [51] Wikipedia contributors. *44th G7 summit — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=44th`G7`summit&oldid=990083488](https://en.wikipedia.org/w/index.php?title=44th%27G7%27summit&oldid=990083488). [Online; accessed 15-December-2020]. 2020.
- [52] Wikipedia contributors. *46th G7 summit — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=46th`G7`summit&oldid=992486931](https://en.wikipedia.org/w/index.php?title=46th%27G7%27summit&oldid=992486931). [Online; accessed 15-December-2020]. 2020.
- [53] Wikipedia contributors. *Longest word in English — Wikipedia, The Free Encyclopedia*. [Online; accessed 15-December-2020]. 2020. URL: [https://en.wikipedia.org/w/index.php?title=Longest`word`in`English&oldid=993507700](https://en.wikipedia.org/w/index.php?title=Longest%27word%27in%27English&oldid=993507700).
- [54] Wikipedia contributors. *School strike for climate — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=School`strike`for`climate&oldid=993965396](https://en.wikipedia.org/w/index.php?title=School%27strike%27for%27climate&oldid=993965396). [Online; accessed 15-December-2020]. 2020.
- [55] Wikipedia contributors. *United States withdrawal from the Paris Agreement — Wikipedia, The Free Encyclopedia*. [https://en.wikipedia.org/w/index.php?title=United`States`withdrawal`from`the`Paris`Agreement&oldid=994176011](https://en.wikipedia.org/w/index.php?title=United%27States%27withdrawal%27from%27the%27Paris%27Agreement&oldid=994176011). [Online; accessed 15-December-2020]. 2020.
- [56] Savvas Zannettou, Joel Finkelstein, Barry Bradlyn, and Jeremy Blackburn. *A Quantitative Approach to Understanding Online Antisemitism*. 2019. arXiv: 1809.01644 [cs.CY].