INFOMMDI Assignment

Visualizing language biases over time using Reddit comments data.

September 2020

Authors:
Otto Mättas,
Filippo M. Libardi

Supervisor:
Alexandru Telea

I. Introduction

In this paper we present a visualisation system that displays linguistic bias trends over time.

Generally, biases are identified as cognitive constructs that translate in a group of people having mutual ideas about how another group commonly behaves. Language functions as both a reflection and propagation of generalizations that individuals convey with them.
Hence, a meaning of etymological conduct that outcomes from these cognitive constructs is that of linguistic bias, defined in Beukeboom (2014) as 'a systematic asymmetry in word choice as a function of the social category to which the target belongs'.

The work hereby proposed heavily relies on Ferrer et al. (2020). We intend to take the modelling method presented there, train our own machine learning model and additionally display its results over a time-span.

The authors *present a data-driven approach using word embeddings to discover and categorise language biases on the discussion platform Reddit.*

Reddit is a social media platform that is organised in clusters of members (i.e. subreddits). These can be seen as communities of people interested in the same topic. The range of topics is vast and varied, not only restricted to classical notions of similar interests like politics or funny pictures but also specific world views on a very narrow topic. One main characteristic of Reddit is that it is mostly a discussion-based platform.

The aim of our research is to train a model over a specific subreddit's comment corpus, quantify the bias as a function of time and finally visualise its evolution. We assume that given such a visualisation, one could more easily spot and understand the rises and falls of bias as a consequence of real world events. More specifically, an online community's user could better understand why linguistic bias grew in certain points in time by correlating it with events that took place within the same community. From this it stems our research question:

Do associations / correlations rise out of bias and sentiment visualisations after introducing the time dimension?

In section II we present the mentioned previous research paper as well as many papers entailing the field of language bias' identification and discovery. We will draw a line connecting each work to this project's very scope.

In section III we aim to provide an overview of our approach and describe both the model utilised in this project and our own addition to it.

In section IV we propose and classify different visualisation methodologies that we could use for the final data visualisation. Following this, we will carry a discussion on the proposed visualisation methods and their respective strengths and weaknesses.

Furthermore, in section V we describe the data corpus and how we process it in order to extrapolate the needed features. This section will also look at the usage of possible APIs to gather data on-the-go. Certain limitations with previous research are also covered.

In section VI we will describe the validation method adopted and how we apply it. The validation entails mostly describing potential value of our method in the real world applications.

In section VII, section VIII and section IX we will analyse the results obtained and discuss possible improvements of the tool.

Lastly in section X we will present a work plan. This section is only kept there for giving an indication on our scope and purpose during ongoing research. It will be removed in the final stages of our research project.

II. Related Work

Previous literature comprehensively explains that language both reflects and propagates social generalizations. However, such investigations

principally influence human surveys, word reference and subjective examination, or knowledge on various languages. This research project will be heavily based on a recent research Ferrer et al. (2020). Here authors make use of a Word2Vec model to estimate words embeddings and their related bias.

A. Word2Vec

Word2Vec is a natural language processing method, namely a commonly used algorithm. As the name suggests, Word2Vec translates words into a dense vector. The incorporated calculation process utilizes a neural network model to understand word associations from an enormous corpus of text. When trained, a particular model can distinguish interchangeable words or predict extra words for an incomplete sentence. The vectors are evaluated carefully with the end goal that a straightforward numerical capacity (the cosine similarity between the vectors) demonstrates the degree of semantic similitude between the words conveyed by those vectors.

As Ferrer et al. explains in the research repository, the research makes use of the gensim python package to build and train the model.

In addition, they add a layer of abstraction which makes use of the model to estimate bias and most biased words. These concepts will be formally further explained in subsection III-A.

B. Word embeddings

The paper Garg et al. (2018) also presents a computational approach to quantify 100 years of gender and ethnic stereotypes trough Word embeddings. The model of the latter research is trained over 100 years of text data, gathered from Google News and other newspaper data sets. Researchers correlate the trends shown by the model with real life events and perform statistical tests to validate the model. We are using this method to validate our results achieved.

III. Methodology

In this section we aim to extensively describe our approach for building the visualisation tool. We will discuss the approach taken by Ferrer et al. on quantifying the language bias as well as describing what we want to introduce to the proposed method.

More closely, we will describe how the previous method formalises the notion of language bias and we will try to deliver the intuition behind the authors choices in subsection III-A.

In subsection III-C, we add detail to our contribution and explain how we thought to approach its development.

A. Quantifying language bias

The previous model proposed by our predecessors tries to capture the language bias by training a NLP model on a given corpus of Reddit comments. The model learns the corpus' embeddings that transform text into high-dimensional dense vectors and captures semantic relations between words.

When the words are turned into vectors, it is possible to manipulate them as such. For instance, distances between vectors can be evaluated and semantic similarities can be drawn between the concepts that the vectors represent.

A classic example of this comparison is *Queen - Woman ; King - Man*. Intuition suggests that if the vector (i.e embedding) of the word *man* is added to the vector of the word *royal*, it results in the vector of the word *king*.

Similarly, the same process can be applied to the word *woman* which translates to the word *queen*, when the word *royal* is introduced. The visual representation of the latter example is given in Figure 2 (ref).

Once the model has estimated all the embeddings of the corpus, it can be given an additional list of words. This will enable the process to output the most biased words out of the corpus and towards the list of words provided. More details on this procedure will be seen in the following sections.

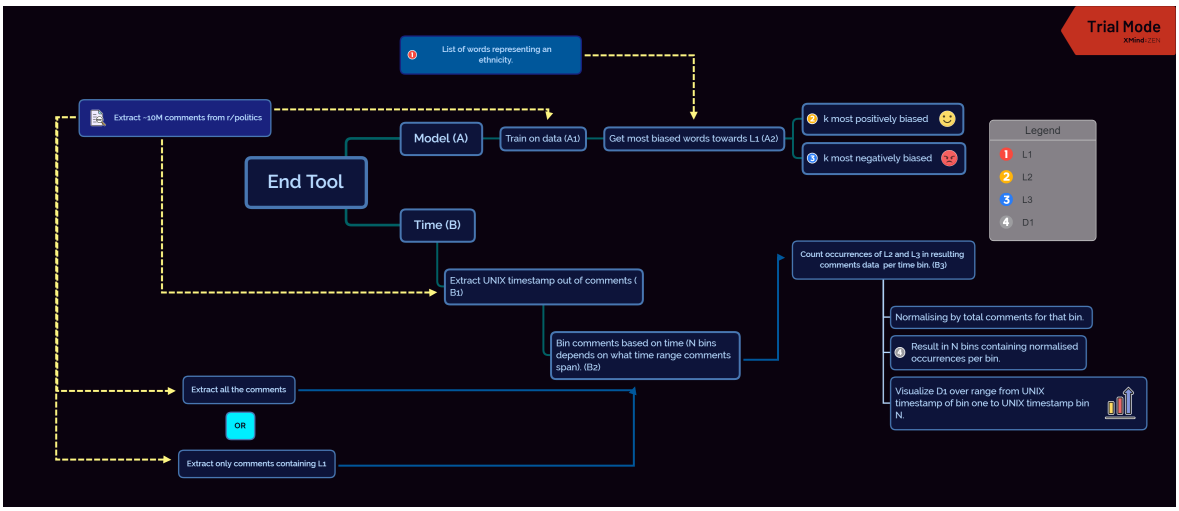

Fig. 1: An overview of the whole pipeline.

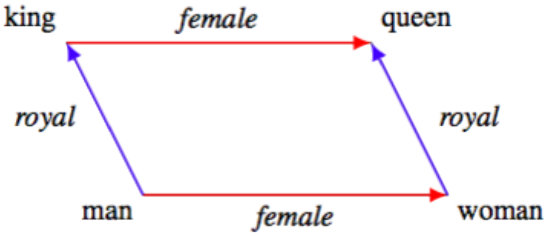

Fig. 2: The visual representation of word embeddings in a vector space (ref)

B. Formalising language bias

Given all the calculated embeddings of a corpus (Reddit comments in our case) and two lists of target words, the model identifies the most biased words towards these lists in the corpus.

Let $S_1 = \{w_i, w_i + 1, ..., w_i + n\}$ and $S_2 = \{w_j, w_j + 1, ..., w_j + n\}$ be sets of target words that try to convey a concept.

For example, S_1 could try to identify the concept of *male* (e.g $\{he, son, his, him, father, male\}$) whereas S_2 could try to identify an opposite concept, such as *female* (e.g $\{she, daughter, her, mother, female\}$).

The model computes the centroids of each target set ($\vec{c}_1$ for S_1 and $\vec{c}_2$ for S_2) obtained by averaging the embedding vectors of word $w \in S$. A word belonging to w is biased towards S_1 with respect to S_2 when the cosine similarity between the embedding of $\vec{w}$ is higher for $\vec{c}_1$ than for $\vec{c}_2$.

$$\text{Bias}\,(w, c_1, c_2) = \cos\left(\vec{w}, \overrightarrow{c_1}\right) - \cos\left(\vec{w}, \overrightarrow{c_2}\right) \quad (1)$$

, where $\cos(u, v) = \frac{u \cdot v}{\|u\|_2 \|v\|_2}$.

Positive values of *Bias* mean that a word w is more biased towards S_1, while negative values of *Bias* mean that w is more biased towards S_2.

Given this definition of bias, the model also estimates the k most biased words towards the given sets. This feature is fundamental towards the building of our end visualisation. More details about this are given in subsubsection III-B1.

1) Most biased words: Let V be the vocabulary of a word embeddings model.

The model identifies the k most biased words towards S_1 with respect to S_2 by ranking the words in the vocabulary V using *Bias* function from Equation 2:

$$\text{MostBiased}\,(V, c_1, c_2) = \arg\max_{w \in V} \text{Bias}\,(w, c_1, c_2) \quad (2)$$

The intuition behind the approach described in subsection III-C is to use the k most biased words in order to quantify bias. The latter translates into counting occurrences of these k over time.

2) Sentiment analysis: When talking about bias, there are often misconceptions and confu-

sion over negative bias, positive bias and how they relate to each other. As bias is essentially a human tendency to associate two concepts with one another, it does not necessarily have positive or negative connotations attached to it.

In order to understand whether the bias is negatively or positively charged, the model makes use of sentiment analysis.

More specifically, the model implements VADER, *A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text*, introduced in Hutto & Gilbert (2015). The model estimates the average of the sentiment over a set of words W as follows:

$$\text{Sent}(W) = \frac{1}{|W|} \sum_{w \in W} SA(w) \tag{3}$$

, where SA returns a value $\in [-1, 1]$ corresponding to the polarity determined by the sentiment analysis system, -1 being strongly negative and $+1$ strongly positive. As such, Equation 3 always returns a value $\in [-1, 1]$.

We will make use of this sentiment analysis to estimate the connotation of the bias found.

C. Adding a time dimension

We found challenges in interpreting the meaning of visualisations presented by our predecessors in their research. Namely, it is hard to understand, what is signified by their graphs, how exactly does the bias evolve in time. Moreover, the practical challenge in our research scope came from actually attaching a time factor to the bias.

While training, the model does not take time as a feature into account. This means we had to find our own way of including time to the model output.

Hereby, we aim to describe our addition to the model proposed. The depiction of our approach is twofold: firstly, we will describe how we aggregate the bias values returned by the model so as to fit into a temporal timeline. Secondly, we show what this timeline entails and what it formally translates to.

An overview of the procedure is shown in Figure 1.

Once the model has returned a list of k most biased words ($A1$ in Figure 1), we take only the words that have a positive or negative connotation (found through sentiment analysis) and that are tagged as adjectives (these are represented as $L1$ and $L2$ respectively in Figure 1), leaving out the biased words that have no polarity.

Then, we iterate through the same comments the model has been trained on and count the occurrences of each word in the list.

A negative and positive *bias score* is accumulated while doing so.

The *bias score* is given by accumulating the sentiment polarity defined in Equation 3. Formally, we define our *bias score* as follows.

$$BiasScore = \frac{\sum_{w \in C} Sent(w)}{|C|} \tag{4}$$

, where C is a a set of all the words present in all the comments of one day.

The aggregated sentiment is normalised by the cardinality of the set (i.e. number of comments) in order not to favor days, where users were commenting more actively than on average.

Because we are iterating trough the comments in their original form, we can access the timestamp attached to them ($B1$ in Figure 1).

As mentioned, we bin the time on a daily basis (one bin signifies one day), where for each bin we sum up all the bias scores and normalise by the total number of comments present in that specific bin ($B3$ in Figure 1), following Equation 4. Once k bins have been evaluated with their respective normalised *bias score*, we can display them over time.

An implementation of the thought approach is shown in Figure 3. The plot shows a list of words that are the most negatively biased towards the Mexican people. Because the bias considered in this case only has a negative connotation (i.e. frequency of insults correlated/biased towards Mexicans), we see the line plot extending on the negative side of the plane.

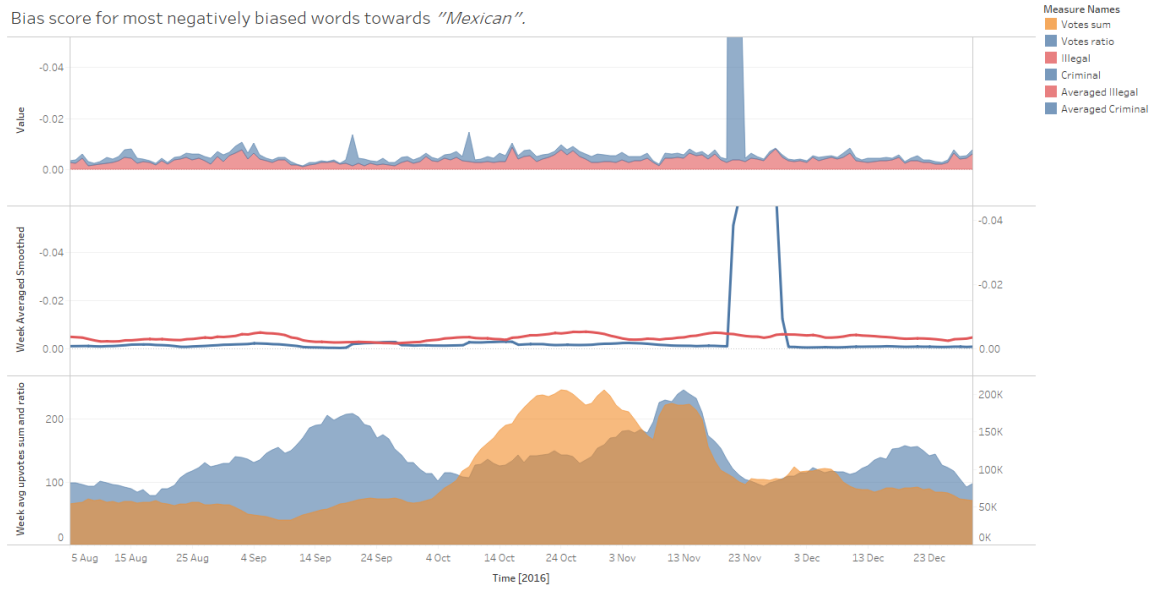

Fig. 3: *Bias Score* towards Mexican ethnicity evaluated by training the model on ~20 million comments of the subreddit r TheDonald. The *Bias Score* is evaluated as shown in Equation 4. The middle graph introduces a smoothed view by averaging over the value of a week for bias score. The bottom one shows the accumulated score of the day (upvotes + downvotes) as well as the ratio between negative and positive comments (of positive / of negative).

An original approach suggests the end tool should be able to comprehend a visualisation for both the negative and positive bias for the word lists given to the model. Following this, we will discuss how to approach and visualise this in section IV.

The problem with the latter approach is that the positive and negative bias often have a major discrepancy in the order of magnitude they are expressed in. This means that the negative *bias score* has much greater values than the positive one. Accordingly, we finally decided to only visualise the negative bias towards a related concept for the purposes of our research.

1) Subsampling: An additional note is to be made on the size of the data corpus. As described in subsection V-A, the data we retrieved extends into gigabytes in size. This introduced a novel issue, which is not only related to the model implementation but also to training time and the robustness of the training method itself.

After realising that the time needed for training a model on such a corpus largely exceeded the time available to us, we decided to subsample from this set of comments, in order to reduce the size while still maintaining the meaningfulness of the data.

There are several possible ways of subsampling from a distribution, we chose simple random sampling and the reason is twofold.

Firstly, as mentioned, the data at hand did not make it easy to perform any sort of operation like dividing into subgroups or performing an analysis of the word distribution. Only running line by line creates significant computational load and increases processing time. Secondly, random sampling has been proven to be accurate in previous research and the process has shown to maintain the essence of the data, conveyed in the end. Taherdoost (2016) West (2016)

A formalisation of the methodology applied is described below in Equation 5.

$$\begin{aligned} P &= 1 - \frac{N-1}{N} \cdot \frac{N-2}{N-1} \cdot \cdots \cdot \frac{N-n}{N-(n-1)} \\ &\overset{\text{Canceling:}}{=} 1 - \frac{N-n}{N} \\ &= \frac{n}{N} \end{aligned} \tag{5}$$

, where N is the total number of data points in the set (~482031200 in our case) and n is the sampled population (~4820312). This means we are sampling around 1% of the total population

or user base in the given subreddit to be more specific.

After reducing the population size, we applied the method to extract the *bias score* explained in Equation 4 and exported everything into a final .csv file, only containing two columns (the date and the bias score of a particular comment).

2) Votes ratio and sum: Lastly, one additional action on the data has been performed.

Reddit is known for its upvoting and downvoting system, where users can not only show their support towards (upvote) a comment or post, but also express a negative feeling about it (downvote). The final score of a comment or post is the sum of both down- and upvotes given to it.

We tried to understand if the bias score's rise and fall over time might correlate to people having a more active discussion on the platform, overall. The *activity* in this case translates to controversiality - that is, posts with both down- and upvoted comments attached to them.

Logically, a post will be more controversial, if it has a great number of negative and positive comments. This is captured by the *votes ratio* (blue area in bottom graph of Figure 3), and is expressed from $|positivecomments|/|negativecomments|$.

Additionally, we display the overall sum of votes in a day, this is to see if users prefer days where a great amount of *bias score* is present (orange area in bottom graph of Figure 3).

These metrics can be seen to validate our intuition. Both, the overall sum of votes and the ratio go down when there is a spike in usage of biased words (in the chart, this can be seen around November 2016).

Essentially, looking at the ratio of negative comments versus positive comments means, that there were comments that people from the subreddit did not agree on, sparking discussions.

Consequently, this suggests that when users engage in debates with other users, their language becomes more biased. Whereas when all users are agreeing on what is said the language can be described as "normally" biased or biased as it would regularly be for each user.

This factor could not be seen without a temporal addition to the previous research. We retain it an interesting and significant finding, proving our method holds ecological validity.

IV. Visualization

In the following sections we would like to discuss visualization methods taken into account.

A. Design

We are trying to represent data that expands over time. Also, the data has a meaningful relationship with zero. The two designs considered for our visualisations are an Area Chart and a Normalised Area Chart (mock-ups of these are shown in Figure 4, (ref)).

The idea is to have positive and negative bias expanding to both sides of the Cartesian Plane on the y-axis. This goes along the intuition of negative bias being negatively quantified and positive bias being positively quantified. Colors will more likely represent words and will be stacked one onto the other so to give an idea of which word is the "major contributor". Before stacking them up, words' bias value will be sorted by area size (according to the design mocked in Figure 4, (ref)).

All the visualization considered are displayed on this publicly accessible Tableau Public Page.

1) Run Chart: This kind of visualisation is often used to represent data that evolves over time. Each data point is plotted and then connected with a line. This also makes it form a line plot. This type of visualisation would suit our needs in principle. From the more aesthetic perspective, we thought it would be more appealing to highlight the area under the curve. This also emphasizes the magnitude of

the bias contribution of each word.

2) Stacked Area Chart: An area chart is basically the same as the previously mentioned Run Chart. On both, area charts and line graphs, data points are plotted and afterwards connected by a line to show the estimation of an amount at each point in time. Area charts are not quite the same as line charts as the zone between the x-axis and the line is filled in with shading or color. Additionally, we aim to stack each variable onto each other, so to make it more clear what word has had the most contribution at each time step with respect to all the other words.

3) Normalised Stacked Area Chart: A variation of the stacked area chart is the percent (or normalised) stacked area chart. It is essentially the same but data points are normalised at each time stamp. That allows the viewer to study the percentage of each variable in relation to the whole more efficiently.

B. Implementation

We will be using Tableau Desktop (ref) to visualise the data and export it in a dashboard on the provided public cloud storage solution. As mentioned, an implementation of each of the visualisation methods considered has already been implemented and is publicly available at this Tableau Public Page.

The Tableau visualisation software rendered making the final implementation extremely easy. No preparation of the data was needed on the Tableau side. Before feeding it to the model, only pre-processing of the corpus was performed.

After seeing the result of each individual visualisation (all published on this Tableau Public Page), an analysis of the more insightful (while still visually pleasing) methodology has been performed.

Firstly, the main difference between the run chart and stacked area chart is that there are highlights under the curve and the values are stacked onto each other. This gives a more valuable first visual aid in identifying the magnitude of the contribution each displayed value takes to the final outcome.

We implemented both, the run chart (for the smoothed version) and the stacked area chart. The only visualisation that has been excluded is the normalised area chart. The main issue with this visualisation is that our normalisation method does not have common minimum and maximum of all the normalised values.

More accurately, we normalise each day's *bias score* by the number of comments posted during that day. This results in each day having its own minimum and maximum values and if we were to display this in a normalised area chart, the percentage towards each day contributed would not add up to 100. This could create confusion in the viewer.

If we were to normalise by the greatest number of comments overall, then we could plot the area chart in a normalised fashion, being sure that each data point contributes equally to form the final area, leaving no blank space.

V. The data

As researchers before us, we also chose for social news aggregation, web content rating, and discussion website Reddit. As the company's motto says, it is aiming to be "the front page of the internet" which also coincides with our goal of having ecological validity and real world correlations.

In principle, we are using open data APIs as we are investigating the comments shared publicly on internet.

A. Limitations from previous research

As we have investigated data collection methods from previous research and found it lacking, this section will see an expansion with sharing the weak backdrop.

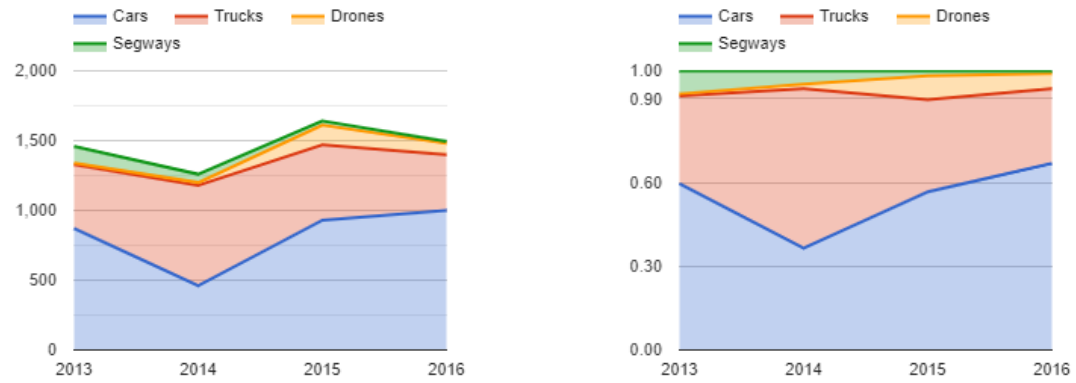

Fig. 4: The visualisation of the two considered designs. On the left the Area Chart and on the right the Normalised version. (ref).

```
[ec2-user@ip-172-31-12-151 LanguageBiasesInReddit]$ python3 Run.py
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/ec2-user/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
[nltk_data] Downloading package vader_lexicon to
[nltk_data]     /home/ec2-user/nltk_data...

        ********************************************************************************
        Test run using a toy dataset of 1000 comments collected from r/TheRedPill
        ********************************************************************************

Training new model Datasets/toy_1000_trp.csv
->Starting with Datasets/toy_1000_trp.csv [body], output Models/toy_trp_model, window 4, minf 10, epochs 5, ndim 200
Traceback (most recent call last):
  File "pandas/_libs/parsers.pyx", line 1149, in pandas._libs.parsers.TextReader._convert_tokens
  File "pandas/_libs/parsers.pyx", line 1279, in pandas._libs.parsers.TextReader._convert_with_dtype
  File "pandas/_libs/parsers.pyx", line 1295, in pandas._libs.parsers.TextReader._string_convert
  File "pandas/_libs/parsers.pyx", line 1518, in pandas._libs.parsers._string_box_utf8
UnicodeDecodeError: 'utf-8' codec can't decode byte 0xbf in position 1: invalid start byte
```

Fig. 5: Formatting error in the original algorithm.

The main issue we found is that the experiment from the previous researchers carried out is not directly replicable.

Mainly, this is due to the code they provide not being able to handle data with size greater than their initial data set. Additionally, they do not provide complete details on technical specifications of the machine the experiment has been run on. After approaching the researchers, we finally received an indication of the technical specification. xfold (2021)

This confirmed our worries about replicating their work - the previous implementation is only optimised for the data set they used originally.

Additionally, previous research did not specify the exact data set they used. Furthermore, the sample data set presented in the repository does not reflect the actual data set. When trying to run the experiment without changing anything in the original code or the data, the implementation fails. Error seems to reflect on the sample data's formatting which can be worked around but should not.

As to find clarity about the initial data set used, we resulted to going back another step.

As there was a clear reference to another research paper about collecting the data from Reddit, we investigated that as well. Baumgartner et al. (2020)

Unfortunately, it focuses on collecting the data overall. Also, no specific details about their technical implementation are given. If given, it would give a hint as to how the follow-up research on the language bias was conducted, though. We got many technical tips about the Pushshift platform and how it is built on a high level. Again, the details are missing.

We approached the researchers who built the Pushshift platform to ask for more details but they have not responded to our outreach at the time current research was comprised and finalised.

It was clear that the researchers had used Pushshift API for their data collection, though.

This originally made us think that we could train the model on two full data sets, respectively of 5 and 20 gigabytes in size. The action resulted in several different issues, mainly the code was not optimised to handle that size of data and so the RAM of our computing clusters ran out quickly even just during the preprocessing steps.

Additionally, the training itself did not implement methodologies that the python package they used provides for training on large datasets

(i.e. streaming from a file directly rather than storing in a variable).

Many changes in code have been implemented and the whole pipeline has been refactored to comply with a large data, or at least larger than the initial data set.

B. Reddit API

At first, we investigated Pushshift API which was also mentioned in previous research Ferrer et al. (2020). As they have also referenced it from related research Baumgartner et al. (2020), we investigated the underlying ingestion method. Unfortunately, it was not disclosed to any degree so we had to rely on our own skills and expertise in collecting information from publicly available APIs.

Developed by third-party developers, Pushshift API has abstracted away from the official Reddit APIs and become more user-friendly to source linked data from with little effort. Unfortunately, the API is not very reliable as we receive many timeout errors while being connected for ingestion.

Additionally, they publish static data sets from archived content to be served and used freely. Unfortunately again, here the technical issues are even more visible as the data sets are stored in static files up to the size of 15GB which need to be downloaded in one continuous network session. The service is not reliable for accessing data sets larger than a few megabytes, though. From error responses received, the service does not seem to be running on scalable infrastructure that is able to support taxing requests like ours.

We moved on to investigating the PRAW Python framework for accessing the Reddit API directly.

PRAW, an acronym for "Python Reddit API Wrapper", is a Python package that allows for simple access to Reddit's API. PRAW aims to be easy to use and internally follows all of Reddit's API rules. With PRAW there's no need to introduce sleep calls in your code. We need to give our client an appropriate user agent and we able to start querying directly.

Right now, we are still waiting on PRAW developers to respond to specific questions about the framework in order to ingest our archive properly. To mitigate the need for data, we have started ingesting newly created comments since 20/12/2020 with a valid PRAW method.

We moved on to investigating the official Reddit API. Even though not meant for high-volume data aggregation and ingestion, the API is closest to the source of truth. Also, there are many specific questions that the Reddit developers are best suited to answer in the first place. We are also waiting on Reddit developers to respond to specific questions about the framework in order to ingest our archive properly.

C. Format

We are ingesting raw JSON comment data which will be transformed during ingestion. The ingestion process results in a CSV-formatted file, including the comment body and timestamp which then will be sent to the machine learning process explained in section section III. The statistical model previously built, plus our addition of the time dimension will output yet another CSV-formatted file, now only including the bias score (per word per bin) and time. Final data, including the comment body, timestamp and bias score will be forwarded to the visualisation process.

In general, hold to web standards and follow best practices in our formatting.

D. Subreddits

We are aiming to make an informed selection and choose for socially relevant and active communities. We are looking for ones that will be representative or with an user base, which is ideologically evenly distributed over the topics discussed. Also, activity is taken into consideration.

To start, we sought out preliminary statistics on the content of some political subreddit's comments. Namely, we are interested in race and bias towards it. This comes from the authors' intuition on what topic could be considered relevant in the current world. To sample, we are using official racial and ethnic categories and definitions for National Institutes of Health (NIH) diversity programs. We count all the mentions of races (noun form) in the subreddits focused on US politics.

1) Race and ethnicity: The 2010 US Census included changes designed to more clearly distinguish Hispanic ethnicity as not being a race.[?] As we are looking at racial bias, we ought to exclude the ethnicity data but think keeping it allows to avoid some confusions with our audience. We are defining and using for our purposes the following:

- Hispanic or Latino. A person of Cuban, Mexican, Puerto Rican, South or Central American, or other Spanish culture or origin, regardless of race. The term, "Spanish origin," can be used in addition to "Hispanic or Latino."
- American Indian or Alaska Native. A person having origins in any of the original peoples of North and South America (including Central America), and who maintains tribal affiliation or community attachment.
- Asian. A person having origins in any of the original peoples of the Far East, Southeast Asia, or the Indian subcontinent including, for example, Cambodia, China, India, Japan, Korea, Malaysia, Pakistan, the Philippine Islands, Thailand, and Vietnam.
- Black or African American. A person having origins in any of the black racial groups of Africa. Terms such as "Haitian" or "Negro" can be used in addition to "Black or African American."
- Native Hawaiian or Other Pacific Islander. A person having origins in any of the original peoples of Hawaii, Guam, Samoa, or other Pacific Islands.
- White. A person having origins in any of the original peoples of Europe, the Middle East, or North Africa.

2) Political subreddits: The authors made a preliminary selection from known regional subreddits for sampling:

- r\Politics
- r\US Politics
- r\American Politics
- r\Canada
- r\Europe
- r\American Government
- r\UK Politics
- r\Euro
- r\Palestine
- r\EU Politics
- r\Middle East News
- r\Israel
- r\India
- r\Pakistan
- r\Cascadia
- r\Iran

3) Subreddit sampling: This is the list of words we are searching for in the comments for any subreddit:

- hispanic, latino, spanish origin
- american indian, alaska native
- asian
- black, african american, haitian, negro
- native hawaiian, other pacific islander
- white

E. API endpoints and queries

We planned for harvesting data from selected subreddits via APIs. We started by using Pushshift API, then moved to Reddit APIs via different methods. As we found constraints with them, namely on real-time data processing and platform reliability, we looked for static data instead.

F. Data ingestion

Finally, after many failed attempts to gather our own data set via the official Reddit API and via the Pushshift API, we discovered a

static data set online which did not include all comments from the decided subreddit r/politics but held most comments. Namely, we found data from the inception of the subreddit until the end of 2018.

We decided to move forward with the incomplete data set and limit the scope of our research based on it. We could focus our efforts to the time before 2019 to work on proving our idea. Data from the start up to 2018 has been made available by Cornell University researchers for their own research purposes. We were able to find also r/politics among the published data.Jonathan P. Chang (n.d.)

G. Preprocessing

Trying to run the original algorithm on our data set failed. Mainly, the script was unoptimised for data other than the original authors used. Most often, the failure came from memory errors. The algorithm was not only for training the model but also for pre-processing the data for training the model.

Unfortunately, the data set size exceeded what the original method was designed and developed for. We used Amazon AWS cloud resources, namely EC2 on-demand instances to run our computational workloads. We extended the technical specifications from 8 CPU cores / 16GB of RAM per instance up to 32 CPU cores / 72 GB of RAM per instance. Unfortunately, the original algorithm did not even get through the pre-processing part of the method without exhausting the memory available. We decided to break off the pre-processing step from the original method in order to keep the cost under control, hoping for a success. From certain thresholds, EC2 instances cost rises while it was not clear if we ever would be able to fit our data set into the memory with the original method.Amazon Web Services (n.d.)

1) Actions: As to give exact technical details of our implementation, we are going to outline all the actions taken in order to reach the final conclusive solution:

1) As the data set came in the form of one JSON file and was too large (20 gigabytes) in size to be worked on as a whole, we split it up in 100 pieces.
2) That allowed us to convert the JSON files into CSV files which is the de facto input format for the training part of the original algorithm.
3) As conversion also reduced the file size remarkably, we decided to merge the 100 CSV files back together in order to start training on the data. In case you wish to replicate this process, do not forget to keep only one CSV header instead of keeping all the headers from all the files. Keeping them could affect the training results as unnecessary noise is introduced.
4) We then uploaded the final CSV to cloud storage for safekeeping and as an accessible source location.

2) Additional findings: As to give more details on tried but failed approaches:

1) We tried exploring the AWS Sagemaker Studio which comes with more tools specific to training models on big source data. Also, it comes with built in algorithms, one of which was Word2Vec, also employed within the original algorithm in previous research. Unfortunately, the original algorithm is still very much a developmental version in terms of clarity of the code. For example, no comments were present that would have helped us directly carry over the core of the method to AWS Sagemaker Studio. Even when we understood how the original method worked in principle, we could not replicate the whole algorithm as there were no clear "building blocks" within the original method to be recreated outside of the method.
2) We soon abandoned the idea of continuing on AWS for the purpose of our project. It is still worth mentioning as the AWS envi-

ronment is for example able to scale as the need for resources during training arises. It is also able to scale back the resources when not needed anymore. What's more - you can cut costs with AWS Sagemaker by using only spot instances which become available at random times during the day so the platform takes care of running the workloads for you.

3) We resulted to running the full script on a hefty 36 core CPU / 72 GB of RAM server. Unfortunately, this also failed as we already explained in section (?)

VI. Validation

The visualisation will be empirically validated by observing the trend of the *bias score* and try to infer causal relationship of spikes with real life events.

VII. Results

We will develop more details about this section in week 2, according to our plan explained in section X. Overall, we aim for statistical significance to help bias research along, as well as provide clear ecological validity for our research.

VIII. Discussion

After finalising analysis on results, we move on to discussing the implications. This section will be clarified on week 3 as explained in section X. Overall, we aim for sparking ideas also new to our team.

A. Limitations

B. Future research

There are many such sites with APIs available so our implementation might be duplicated to analyse other websites after Reddit too.

IX. Conclusion

This section will be clarified on week 3. We are planned for making summaries and finishing the report as explained in section X.

X. Timing

In this section we will provide a schedule of the project's development - of course we intend to respect it as much as possible. A more detailed overview of the work plan listed below is shown in Figure 6

- 7-13 December: Visualising the data over a simple line-plot. Familiarising with the API.
- 14-20 December: Familiarising with Tableau. Testing a bigger data-set. Researching possible visualisation design (consider feedback).
- 21-27 December: Selecting subreddits and contact moderators. Testing the selected visualisation design.
- 28-3 December/January: Harvesting data from selected subreddits and importing it into Tableau. Writing a form to send to moderators.
- 4-10 January: Working on Tableau visualisation. Sending moderators the form.
- 11-17 January: Preparing the presentation. Collecting and analysing the results.
- 18-24 January: Polishing report.
- 24-27 January: Hand in week.

Week	Date	Time	Activity	Comments	Comments	Deadline
49	04-Dec		Work on timeline	Gather data about course deadlines etc	Otto	10:30
	04-Dec		Agree on timeline	Timeline set up	Filippo/Otto	
	04-Dec	10:30	Weekly Friday Meeting	Read through previous research	Filippo/Otto	
	04-Dec		Work on baseline implementation	Read through previous research	Filippo	
			Work on API data colletion	Read through previous research	Otto	
50	07-Dec	10:30	Weekly Monday Meeting	Implement previous research technical solution / API baseline	Filippo/Otto	
		11:00	Supervisor meeting	Baseline implementation / API data collection overview	Filippo/Otto	
	08-Dec	18:00	Send project draft for revision (Alex)	Gather feedback on the problem-solution combo during meeting	Filippo/Otto	
	09-Dec		Edit draft based on feedback	Carry over core improvements / Improve overall look	Filippo/Otto	
	09-Dec		Finish project draft	Confirm all the changes with final read-through	Filippo/Otto	
	09-Dec		Hand in final project draft	Finish project draft		18:00
				GOALS this week:		
				- Visualising the data over a simple line-plot.		
	11-Dec	10:30	Weekly Friday Meeting	- Familiarising with the API.		
				GOALS this week:		
				- Familiarising with Tableau.		
				- Testing a bigger data-set.		
51	14-Dec	10:30	Weekly Monday Meeting	- Researching possible visualisation design(consider feedback).		
	16-Dec	15:00	Class			
	18-Dec	10:30	Weekly Friday Meeting			
				GOALS this week:		
				- Selecting subreddits and contacting moderators.		
52	21-Dec	10:30	Weekly Monday Meeting	- Testing the selected visualisation design.		
	~~23-Dec~~	~~15:00~~	~~Class~~			
	25-Dec	10:30	Weekly Friday Meeting			
				GOALS this week:		
				- Harvesting data from selected subreddits and modelling.		
				- Importing results into Tableau.		
53	28-Dec	10:30	Weekly Monday Meeting	- Writing a form to send to moderators.		
	~~30-Dec~~	~~15:00~~	~~Class~~			
	01-Jan	10:30	Weekly Friday Meeting			
				GOALS this week:		
				- Working on Tableau visualisation.		
1	04-Jan	10:30	Weekly Monday Meeting	- Sending moderators the form.		
	06-Jan	15:00	Class			
	08-Jan	10:30	Weekly Friday Meeting			
				GOALS this week:		
				- Preparing the presentation.		
2	11-Jan	10:30	Weekly Monday Meeting	- Collecting and analysing the results.		
	13-Jan	15:00	Class			
	15-Jan	10:30	Weekly Friday Meeting			
				GOALS this week:		
3	18-Jan	10:30	Weekly Monday Meeting	- Finalising the report.		
	20-Jan	15:00	Class			
	22-Jan	10:30	Weekly Friday Meeting			
				GOALS this week:		
				- Hand-in week starts.		
4	25-Jan	10:30	Weekly Monday Meeting	- Creating and rehearsing the presentation.		
	27-Jan	14:00	Presentations in class	Project Presentation		
	29-Jan	10:30	Weekly Friday Meeting			
				GOALS this week:		
5	01-Feb	10:30	Weekly Monday Meeting	- Hand-in week continues.		
	03-Feb		Hand in assignment	Project Report Submission		?
	12-Feb		CELEBRATE			

Fig. 6: Detailed work plan.

References

Amazon Web Services, I. (n.d.), *Amazon EC2 On-Demand Pricing*.

Baumgartner, J., Zannettou, S., Keegan, B., Squire, M. & Blackburn, J. (2020), 'The pushshift reddit dataset', p. 10.

Beukeboom, C. (2014), *Mechanisms of linguistic bias: How words reflect and maintain stereotypic expectancies.*, pp. 313–330.

Ferrer, X., van Nuenen, T., Such, J. M. & Criado, N. (2020), Discovering and categorising language biases in reddit, *in* 'International AAAI Conference on Web and Social Media (ICWSM 2021) (forthcoming)'.

Garg, N., Schiebinger, L., Jurafsky, D. & Zou, J. (2018), 'Word embeddings quantify 100 years of gender and ethnic stereotypes', **115**(16), E3635–E3644.
URL: *http://www.pnas.org/lookup/doi/10.1073/pnas.1720347115*

Hutto, C. & Gilbert, E. (2015), Vader: A parsimonious rule-based model for sentiment analysis of social media text.

Jonathan P. Chang, Caleb Chiam, L. F. A. W. J. Z. C. D.-N.-M. (n.d.), *ConvoKit: A Toolkit for the Analysis of Conversations*.

Taherdoost, H. (2016), 'Sampling methods in research methodology; how to choose a sampling technique for research', *International Journal of Academic Research in Management* **5**, 18–27.

West, P. (2016), 'Simple random sampling of individual items in the absence of a sampling frame that lists the individuals', *New Zealand Journal of Forestry Science* **46**.

xfold (2021), *Issue: What runtime parameters are required?*
URL: *https://github.com/xfold/LanguageBiasesInReddit/issues/1*