Tracking Xenophobic Terminology on Twitter Using NLP

Harper Lyon Nick Webb (Advisor)

March 17, 2022

Contents

List of Figures

List of Tables

1 Introduction

Xenophobia refers to a dislike or distrust of foreigners qua their status as foreigners. This dislike has significant overlap with, and may be caused by or cause, other types of bias, like racism and religious intolerance, but it has its own character, expressed most clearly by the vocabulary and style particular to xenophobic speech. Having a dedicated list of words and phrases known to be common in xenophobic posts online would solve one of the main challenges of identifying such speech [4]. Keeping track of new language is itself a challenge, especially given the rapid rate of change in online language and computational tools may be necessary to keep up. For this reason my thesis focuses on developing and evaluating techniques for identifying previously unnoticed xenophobic language among xenophobic tweets using Natural Language Processing (NLP).

1.1 Goal

The high level goal of this project was to develop techniques for identifying the emergence of new xenophobic terminology on twitter. For example, if given a selection of tweets around from the beginning of the Covid-19 pandemic, I want to be able to computationally identify terms like 'kung-flu', 'chinese virus', and 'wuhan flu' as not just xenophobic, but both newly important and xenophobic. Practically, my goal was to create a dataset of tweets discussing issues of immigration or foreign policy and then look for xenophobia (or at least new language) in that dataset.

2 Data

Almost the entirety of the the fall term was spent gathering tweets to analyze, but it became clear by the beginning of the winter that without significant funding or many months of time collecting a sufficiently large dataset distributed over a long enough timescale would be impossible. Fortunately I was able to find a suitable pre-existing dataset available for research use, a collection of tweets about the 2016 United States Presidential election. [2]

2.1 Dataset Description

This dataset was collected by Alex Filatov between September 2016 and February 2017 (inclusive) for the purposes of displaying tweets on a personal website along with some simple analysis, primarily sentiment scores. It contains 61 million tweets which directly tag (i.e. contain @therealdonaldtrump) one of four people deemed relevant to the election: Donald Trump, Hillary Clinton, Bernie Sanders, and Barack Obama.

This is obviously a very broad dataset, both in size and in potential topics: not every tweet which tags a politician is going to be about issues relevant to xenophobia. The first step was therefore to pull out a sufficient quantity of tweets about immigration: essentially producing the dataset I was unable to gather myself out of the larger collection of tweets.

2.2 Filtering

The large size of the Filatov dataset gave me some flexibility in choosing a filtering method. Often when organizing data there is conflict between two goals: to have as much data as possible and to ensure that all data is relevant to the analysis. This dynamic exists here as well; the more picky we are in choosing tweets the fewer tweets end up in the final dataset. However, because I started with such a large dataset I can afford to be quite picky and focus entirely on ensuring the relevance of each tweet.

To this end I used the simplest possible filtering method, selecting only tweets with the keywords 'immigrant' and 'immigration'. This ensures that every tweet selected is, in a very literal sense, talking about immigration. This filter takes the dataset from 61 million tweets to 233 thousand, which is a dramatic drop. This was not entirely unexpected; the relationship between accuracy and data size outlined above would suggest that in adopting such an extreme filter would result in a small dataset. We might have expected more tweets to make it through however, so it is useful to keep in mind that there are many relevant tweets that remain in un-examined and that this will have an impact on any analysis.

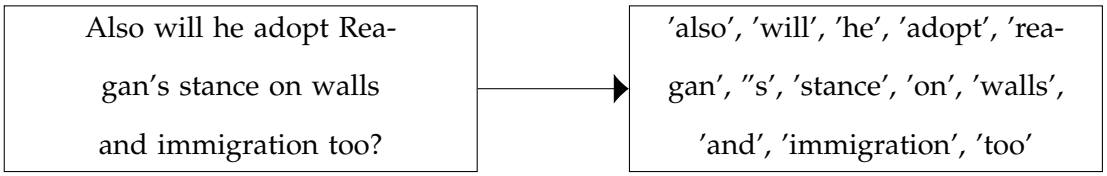
3 Analysis

The bulk of this project has been attempting various methods of identifying new language in a subset of the Filatov dataset. All analysis was performed in python, using the Natural Language ToolKit (NLTK) for essentially all NLP operations. As you will see, the analysis techniques grew naturally from simple attempts to find frequent phrases in the tweet text as I attempted to address holes in previous methods. We begin, however, with some necessary steps to prepare the tweets for essentially any NLP analysis.

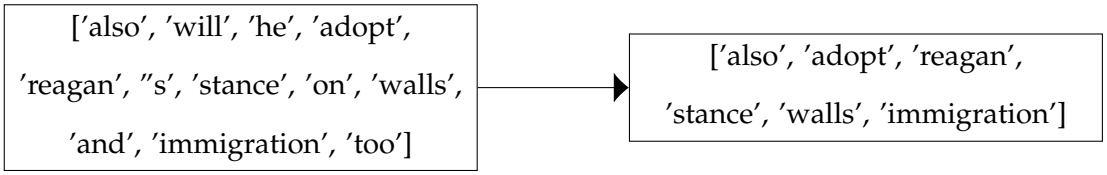
3.1 Initial Processing

Post filtering the tweets still need to be processed into a form conducive for NLP analysis, which was standardized for all experiments. Text was passed through four layers of standard NLP prepossessing: tokenization, stop word removal, part of speech tagging, and stemming.

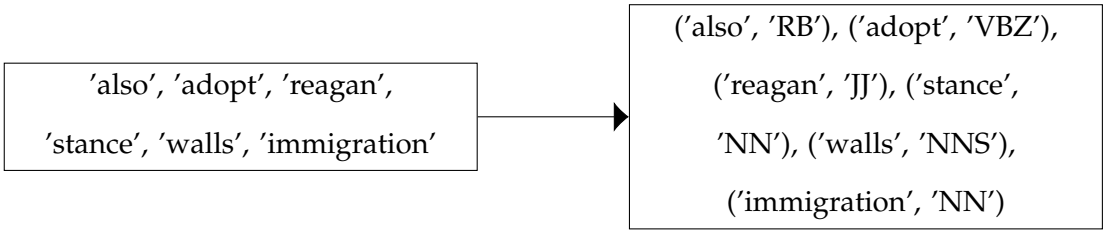
1. **Tokenization**: Tokenization is the process of turning continous text into tokens, discrete pieces representing either individual words or phrases which can then be treated as atomic components of the text. This is one of those tasks that is trivial but tedious for a human but surprisingly tricky for a computer. Tweet text is especially prone to odd structure, so this step is important to get right.



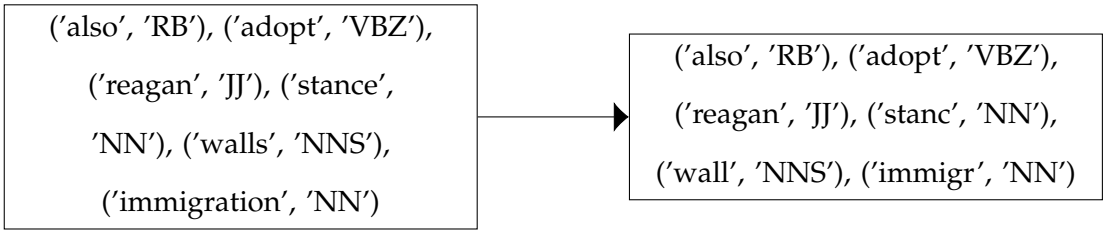
2. **Stop Word Removal**: Once the text has been broken down into tokens we want to remove several things from the text before any serious analysis. This includes what are called stop words: 'to', 'of', 'um', and other words which generally don't contribute to the meaning of the tweet. This is also where we filter out non-words, like URLs and any punctuation not caught in the tokenization step.



3. **POS Tagging**: Once we have a clean, tokenized text we can actually start applying some analysis. The first step is almost always to tag each token with it's part of speech. This is useful information to have in many cases, and helps us distinguish between the same literal word being used in different ways.



4. **Stemming**: The final step of preparing the text is stemming, which is reducing words to their stems. In a sense, while POS tagging is an attempt to identify the same tokens being used in different ways stemming is the opposite: trying to identify different tokens being used in the same way.



The description of these as a set of specific filters is over-simplification of course: each of these processes is a variable that can be adjusted in the course of experimentation, especially POS-tagging and stemming. In fact all experiments were run with and without POS-tagging and stemming, as both have advantages and disadvantages. POS-tagging tends to increase the number of unique tokens in the text (instead of 200 instances of 'immigrant' we now have 15 of ('immigrant', 'NN'), 78 of ('immigrant', 'JJ'), etc). This make some analysis more cumbersome and generally increases the noise. Stemming has, as may be expected, the opposite problem. By collapsing multiple words into one stem we lose information, and we have to be careful not to erase new forms of words (remember that we're looking for new language - it doesn't always help to erase distinct words).

Further, while POS-tags and stemming are very useful analytic tools they can also make results harder for humans to understand. This can be seen in the example above: POS tags turn flagged language from words and phrases to lists of tuples and it can be difficult to understand what

stemmed text originally said. While these aren't really a problem while conducting research, it does make results harder to intuitively evaluate and for this reason we'll only be looking at text for the remainder of this report, unless the tags or stems have particular relevance in an example.

3.2 Frequency Measures

Now that we have all this processed text, what can we actually do with it? The first and most basic thing to do is to measure the frequency of both individual words and phrases in the text. This is a two step process: first we need to generate 'phrases' to measure (since we already have a set of single words), which is accomplished by generating various n-grams for the text. We can then, using again built in NLTK tools, generate a frequency distribution for the text.

Generating n-grams sounds far more impressive than it actually is. An n-gram is simply a sequence of n words, and so generating n-grams for a piece of text is nothing more than taking all sub-sequences of size n. As an example, consider finding all 2-grams of the following sentence:

'How are you today?' $\rightarrow$ ['How are', 'are you', 'you today?']

This is possible for essentially any size, but we'll mostly look at $n \leq 4$, for a few reasons. Practically we want to the keep the size of our n-grams small since as we increase n we dramatically increase the number of phrases we need to store and evaluate. Even at relatively small sizes we can easily run into memory issues, which might be an issue if we were very interested in large n-grams. Fortunately, these start to be less interesting the larger the phrase gets. Very long phrases tend not to repeat, and while a common 2-gram like 'illegal immigrant' might be very important for understanding political speech a 15 word long phrase is probably not a common slogan, nor is it likely to represent an important concept. Basically, we're looking for common phrases or simple word combinations, not common sentences.

With all phrases cataloged we can finally start to identify new language, and the most obvious thing to try is to simply look at the most common words and phrases. We can do this by generating frequency distributions, which is a fancy way of saying we count how many times each token appears in the text. As an example, if we generated a frequency distribution for a single tweet worth of tokens

'trump', 'changing', 'position', 'immigration', 'only', 'position', 'self-centered'

We would end up with a frequency distribution of the form:

trump	1
changing	1
position	2
immigration	1
only	1
self-centered	1

Which is of course completely uninteresting for just the one tweet. When we generate such a table for all of our text the results become more interesting. With such a table in hand for all words and phrases in all tweets we have an immediate potential method for finding new language: we can just look at words or phrases with the highest frequency. Unfortunately, as I identified in the preliminary work for this project, this doesn't work out very well. We tend to identify very familiar language, and learn that when talking about in 2016 people like to say the words 'immigrant', 'illegal', 'trump', and 'immigration', among other classics like 'president' or 'government'. Not all that interesting, and ultimately just a stepping stone towards more sophisticated analyses.

3.3 Linear Trends

As it turns out, simply measuring the frequency of words and phrases doesn't really help us identify new language. The most obvious conceptual problem with the pure frequency approach described above is that it fails to take advantage of any features of our dataset, treating a collection of tweets written over the course of months as if it were a single book. Further, if our goal is to determine change in language over time then ignoring the time component of the dataset is absurd. This observation leads naturally into the next method I tested, which is to produce trend lines for each word or phrase in the text, analyzing the change in usage of that term over time. If we can do this, then identifying new language may just be a matter of looking at the terms which increase most in frequency from the beginning to the end of the dataset.

The first step in this process is to break the tweets up by date and time posted, which involves sorting each tweet into a 'bucket' of tweets posted at a similar time period. There are a number of ways to do this: we could split by real life week, by month, or in some more obscure way like

making adjusting the time span of each bucket to encompass equal number of tweets. This of course becomes another variable to adjust in experimentation, but I found that results tended to be best when the buckets mirrored real life time periods, so monthly and weekly splits worked out well. See Figure 1 demonstrates splitting the data into month long buckets.

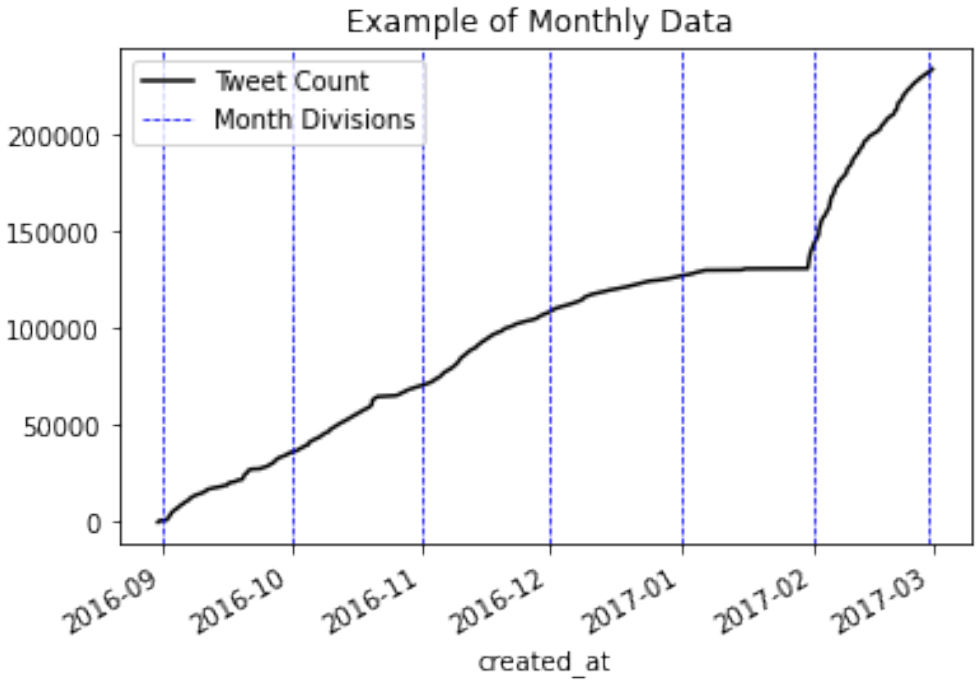


Figure 1: Division of Data into Monthly Buckets.

Note that these collections will not be of equal size, as there were far more tweets selected for some months than others, especially February. For now we'll ignore this, but it will be important soon.

Once we have the buckets set up we can treat them as individual collections of text and perform the same processing and frequency measurements steps as we did to the larger dataset. This results in a frequency distribution for each bucket, so we have, for each word or phrase, the number of times it was used in September, in October, etc. Normalizing these counts by the size of the bucket they come from (it wouldn't be fair to say that a word shows up 100 times in a bucket with 100,000 tweets and only 10 times in a bucket of 100 tweets) we end up with a series of data points for each term, which we can then graph and attempt to identify trends. This is accomplished by running a simple linear regression algorithm on each term, which gives us a linear approximation

Unigram Results	Bigram Results
'realdonaldtrump'	'potus', 'realdonaldtrump'
'immigrants'	'immigrants', 'realdonaldtrump'
'ban'	'realdonaldtrump', 'immigrants'
'potus'	'immigration', 'enforcement'
'president'	'president', 'realdonaldtrump'

Table 1: Unigram and Bigram Results of Linear Trend Analysis

representing the changing frequency of the term over the dataset.

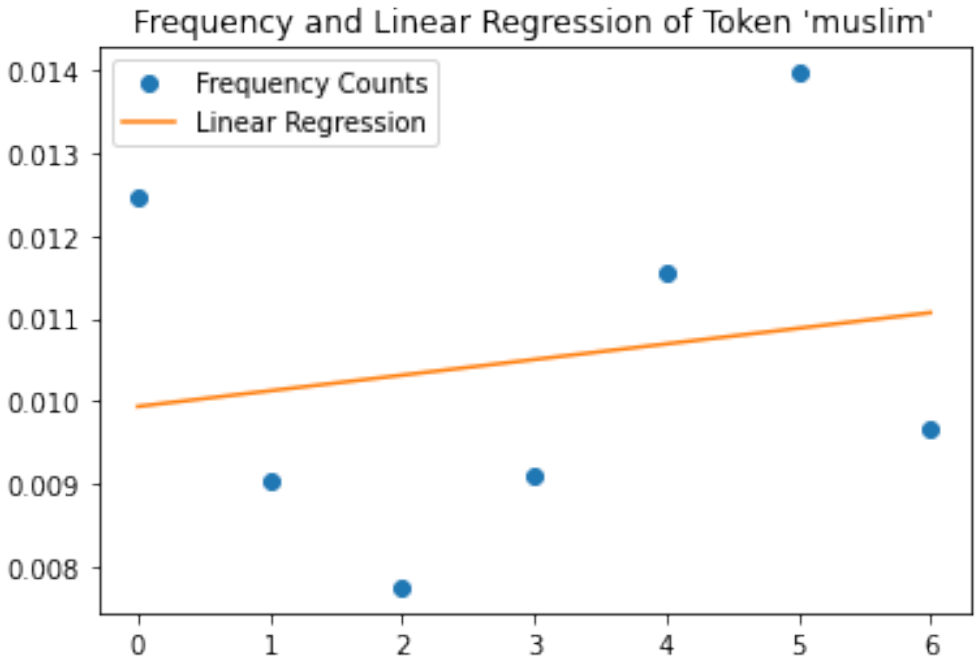


Figure 2: Normalised frequency counts and fitted linear regression for token 'muslim'.

From there, we can look for words with highly positive trend lines (i.e. high slope) to find words which become more common as time progresses and are therefore likely to be new or newly relevant terms. When we do so, we end up with results like those found in the Table 1.

We can make two observations here. First, as 2 makes clear, the linear regressions are bad. Whether this is the fault of my normalization method or an innate feature of the data is not entirely clear, but I suspect that it was a bit of both. Language frequency on social media is, in retrospect, unlikely to follow a straightforward linear pattern. Terminology usage is far more likely to ebb and flow even over short timescales. In other words, the frequency of a terms usage is likely non-linear, so attempting to model it linearly is difficult.

Second, the results are fundamentally uninteresting. There was no new language picked up by this analysis, in fact most flagged terminology consisted of variations on Donald Trump's twitter handle and the words immigrant and immigration (which, recall, were our original selection terms). This is more or less identical to what we would get in a basic frequency measurement. Now, unlike that simple frequency measurement strategy I don't know that we can conclusively say a method like this will never work. There are many possible adjustments to be made here: better normalization, more advanced modeling strategies, etc. It is entirely possible that some similar method exists that would function well, but it will be difficult to find. Because of this difficulty I decided against attempting to perfect this method and to instead try something new.

3.4 Hotspots

Recall that the original goal of the linear trend analysis was to utilize an additional feature of the dataset - the time distribution of tweets - to improve the analysis. While this didn't work out that well, the underlying goal remains strong. This leads to the question: what else can we use? There are a few good candidates, like social information (like counts, retweet counts, who posted the tweets, etc) and the sentiment scores (it might be the case that highly emotional tweets are more likely to contain language), but the one I was most interested in is actually already accidentally present in the linear analysis. Note in Figure 1 that in the month of February we have a large increase in the number of tweets, with almost 25% of the selected tweets being in that month alone. Either our selection process was for some reason biased towards selecting tweets in February, which is unlikely given the simplicity of keyword filtering, or there were many more tweets about immigration in that time period. Assuming it is a genuine spike in immigration tweet volume, this period is very promising. Such spikes are likely to be caused by real world events related in some way to immigration and will be fertile ground for new language to come into use or relevance.

The goal of this final analysis is to examine that period, what I call a Hotspot, to find out what, if anything, is different in that collection of tweets. The general methodology for this was to break the data into buckets, exactly the same as in the linear analysis, but to then search for any bucket with a significantly larger number of tweets than the average. In the case of the Filatov dataset, this was, of course, any bucket covering the month of February. With the hotspots identified, we

Unigram Results	Bigram Results
'immigrants'	immigration', 'ban'
'realdonaldtrump'	'refugees', 'illegal'
'ban'	'immigrants', 'refugees'
'refugees'	'immigration', 'order'
'order'	'executive', 'order'

Table 2: Unigram and Bigram Results of Hotspot Analysis

can then generate the usual frequency distributions and assign each term a rating corresponding to the difference between its frequency in the hotspots and in the rest of the tweets. Theoretically taking the terms with the highest ratings should give us a list of words and phrases which come into more frequent use during this period. Such a list is found in Table 2, and we can note - among the usual inclusion of Donald Trump's username and the original keywords of immigrant and immigration - a common theme of references to refugees, orders, and bans.

If we stop to remember the events of early 2017 this begins to make sense. On January 25th, 2017 Trump issued the first of two high profile executive orders related to federal immigration policy. The second, on the 28th is of more relevance here, as it banned all immigration from many majority Muslim countries, and was heavily criticized for its impact on refugees. [3] It is promising then that we have been able to identify an increase in terms related to these events, as it seems we have correctly flagged newly relevant terminology in the American immigration discourse.

4 Evaluation of Results

Unfortunately, while promising, the results of the hotspot analysis are not entirely what we're looking for. I have determined a method to, at least in this case, identify some newly relevant language, but some problems remain. The constant flagging of known but high frequency terms remains a consistent issue, and even the less obvious identified terminology tends to be unsurprising. For instance, it is interesting to see that refugee becomes more associated with discussions of immigration after the executive order, but its not as if refugee is an entirely new term or just being used to describe immigrants for the first time. I have mostly identified renewed associations, not real evolution in language.

4.1 Future Work

The most promising area for future work would be to improve the filter to used to select tweets. Recall that the filter used to originally gather tweets from the full 61M tweet collection was a simple keyword filter: if a tweet contained the string 'immigrant' or 'immigration' it was selected. It is a very practical solution, implemented because of it's simplicity and left unchanged because of the cumbersome nature of changing the data set. This was a mistake, as I would have been likely to benefit from developing a better filter. A broader filter, even adding additional keywords, would have resulted in a larger dataset, which would have undoubtedly improved the chances of catching new language. It is also possible that in selecting only tweets which literally say immigrant I made it impossible to find tweets which used a different word, losing out on an entire set of new language I would most like to find. The filtering could be improved in many ways, especially by adding more keywords to the list. In fact, one major advantage gained from the hotspot analysis is that we now have several additional keywords to find tweets about immigration, at least in the month of Feburary.

If anyone were to follow directly in the footsteps of this project I would recommend the above improvements, but there are more general research directions that could be followed as well. I would be especially excited about applying similar methods to entirely new datasets. There is nothing unique about either the Filatov dataset, the time period of 2016-17 (simply look at 2020-21), or the topic of immigration. Applying an analysis similar to the hotspot method to similar spikes in tweet volume on any topic should hopefully yield similar results, or at least fail for an interesting reason. In the highly reactive and crisis fuelled social media environment of recent years, such spikes should not be hard to find.

5 Acknowledgements

I would like to thank the Xenophobia Meter Project team at Cornell [1], who sparked my interest in this project and who helped me avoid a few dead ends. I would also like to thank Alex Filatov [2] for his work in compiling the 2016 USA Presidential election tweets dataset and for making it publicly available. Finally I must thank my advisor, Prof. Nick Webb. He is the reason I came to Union in the first place and may very well be the reason I graduate. He has been incredibly

supportive throughout the project and he never once let me lose sight of the real goal of a senior thesis: to challenge myself and learn in doing so.

References

[1] Bao Khan Chau et al. "Tracking Xenophobic Twitter Speech To Inform (and Shift) Policy". In: (2020). URL: https://www.academia.edu/43868921/Xenophobia_Meter_Project.

[2] Alex Filatov. *2016 USA Presidential election tweets*. 2020. URL: https://data.world/alexfilatov/2016-usa-presidential-election-tweets (visited on 03/03/2022).

[3] Reuters Staff. "White House says has updated guidance for green card holders". In: *Reuters* (Feb. 2017). URL: https://www.reuters.com/article/us-usa-trump-immigration-greencard-idUSKBN15G5HB.

[4] William Warner and Julia Hirschberg. "Detecting hate speech on the world wide web". In: June 2012, pp. 19–26.