AMSI
VACATION
RESEARCH
SCHOLARSHIPS
2018-2019

Detection of Australian Racism in Social Networks

Sajit Gurubacharya
Supervised by Dr. Laurence Park
Western Sydney University

Vacation Research Scholarships are funded jointly by the Department of Education and Training and the Australian Mathematical Sciences Institute.

AMSI AUSTRALIAN MATHEMATICAL SCIENCES INSTITUTE

Australian Government
Department of Education and Training

Abstract

Racism is prevalent in our society and also nowadays in social media. It is difficult to detect due to its language dependance and manual methods are infeasible due to the large volumes of data available, therefore automated methods are required. This project aims to examine the utility of specific word embeddings such as word2vec to represent words used in social media in order to classify racism. We analyse problem-specific texts from Australian Twitter users to classify racist words used in the Australian context and visualise models of words using dimensionality reduction techniques such as t-SNE and PCA.

1 Introduction

The rate at which our cities and society as a whole are growing are at an all time high. At the same time, with the advent of the internet and social media, we are more connected than ever. Communication with the opposite side of the world takes places in an instant and ideas and knowledge are much more accessible to the greater public. Online platforms such as YouTube, Facebook, Twitter and others have been leading the way in making this possible.

One key difference between the main form of mass communication we had a few decades ago and now is that now it is taking place through a virtual medium. While this has made communication much easier and cheaper overall, it has also been applicable to more extremist and radical views being shared, one of the main ones being racism. This virtual medium hides our real face and in turn lets us display only what we allow it to. As such, people find it easier to share views which they would not have necessarily done so in person. The growth of hate speech and racist views online would make detecting such sentiments in an automated method beneficial for the whole community.

This project aims to convert words in tweets from Australian Twitter users into vector representations using word embeddings which can be numerically processed in order to classify words that are deemed to be racist in the Australian context, hence allowing for an automated approach. This is a different approach to building a model of such words from analysing generic texts such as from a dictionary or Wikipedia pages because our approach takes into consideration the local dialect, the appropriation of existing vocabulary and the inclusion of more human like speech where racist remarks are used in the relevant context.

Dimensionality reduction techniques are then used to convert higher dimension models of words to lower dimensions for better visualisation of clustering of similar words.

2 Tweets from Australian Users

The dataset for this project used tweets from users who set their location as Australia. There were 14,246 such Twitter users and totalling 25,624,582 tweets, averaging over 1,700 tweets from each user. There were over 250,000 unique words used, but not all of these are necessarily words. They could be representations of unique numbers, acronyms, or even misspelled words. Such non-standard texts were not excluded from the analysis as they are the ones that make the distinction when rather analysing Wikipedia pages.

Common words such as 'the', 'and', 'to' seemed to make up a significant proportion of words used. For example, from a random batch of 389,775 tweets, there were 4,909,917 words of which there were 106,820 (2.2%) occurrences of 'the', 24,979 (0.51%) occurrences of 'be', 85,650 (1.7%) occurrences of 'to', 45,829 (1.0%) occurrences of 'of', 48,314 (1.0%) occurrences of 'and' and 63,096 (1.3%) occurrences of 'a'. These are the six most common words in the English language based on an analysis of the Oxford English Corpus (a collection of texts in the English language, comprising over 2 billion running words). These six words totalled 7.7,% of all words in this batch. This is not surprising as such a distribution has been referred to follow Zipf's law [2] when using a more accurate data set. This also implies that when searching for racist words, which are already fairly rare and used by users on the extreme ends of the distribution, the working dataset for racist words them selves turn out to be significantly smaller than the original data set.

Below, a decreasing pattern can be seen. Some words such as 'be' and 'of' have lower than expected frequency while words such as 'I' and 'you' have greater than expected frequencies. This might be due to tweeting in first person, compared to a dictionary that the ranks were pre-determined from.

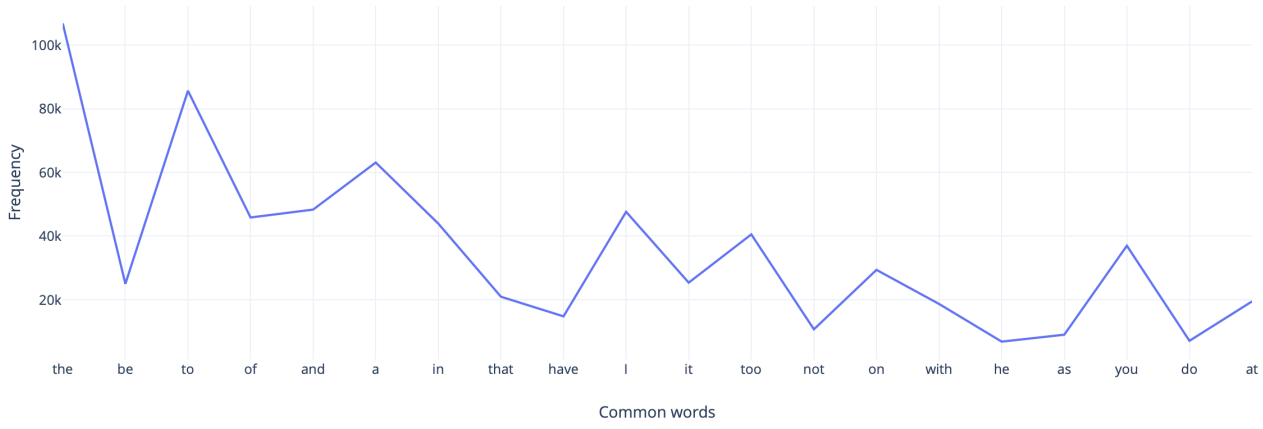

Figure 1: Frequency of the 20 most common words (based on OEC rank) in a batch of tweets

3 Word Embeddings

Word embedding is one of the most popular ways of language modelling techniques in natural language processing [1]. They try to map a word using a dictionary to a vector. This dictionary could be the a dataset of a news archive, Wikipedia pages, or in our case, it is the tweets from Australian Twitter users. The simplest way of such a representation maybe a hot encoded vector where different words are stored in different dimensions. For example, in the sentences "words to vectors" and "words to arrays", the dictionary would consist of ['words', 'to', 'vectors', 'arrays'] and the vector representation of 'vectors' would be [0, 0, 1, 0]. Such representations do not account for similarity between any words. The word 'arrays' would be represented as [0, 0, 0, 1] showing no correlation to 'vectors'.

Newer methods to create word embeddings use the context a word is used in to make sense of the word. In the above sentences, as 'arrays' and 'vectors' were used in a similar context, they would be considered to be similar. Such a method can also recognise tenses of a word such as 'play' vs 'played' and even combination of words such as 'New Zealand' as one word. This is because they tend to be consistently used in a similar context and thus would imply meaning.

3.1 Word2vec

Word2vec is a double layer neural network that are used to produce word embeddings [3]. It was created by a group of researchers led by Tomáš Mikolov at Google. It takes in a large corpus of text and produces a vector space, generally of several hundred dimensions. Similar words in the model have their vectors in close proximity to each other. To find the degree of proximity of vectors, a cosine similarity is calculated. Complete similarity of 1 is expressed as a 0 degree angle, while no similarity of 0 is a 90 degree angle.

$$\cos(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A}.\mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} \mathbf{A}_i \mathbf{B}_i}{\sqrt{\sum_{i=1}^{n} (\mathbf{A}_i)^2} \sqrt{\sum_{i=1}^{n} (\mathbf{B}_i)^2}} \tag{1}$$

The two models used in Word2vec are Continuous Bag of Words (CBOW) and Skip-gram. CBOW uses the context to predict the target word while Skip-gram uses a word to predict a target context. In the sentence "I kicked the ball", if the target word is 'kicked', CBOW takes in the surrounding words and predicts the probability of the word fitting. Skip-gram on the other hand takes in the word and predicts the probability of the surrounding words appearing in its context. The latter method was used in this project as it tends to produce more accurate results on large data sets while also working better with less common words.

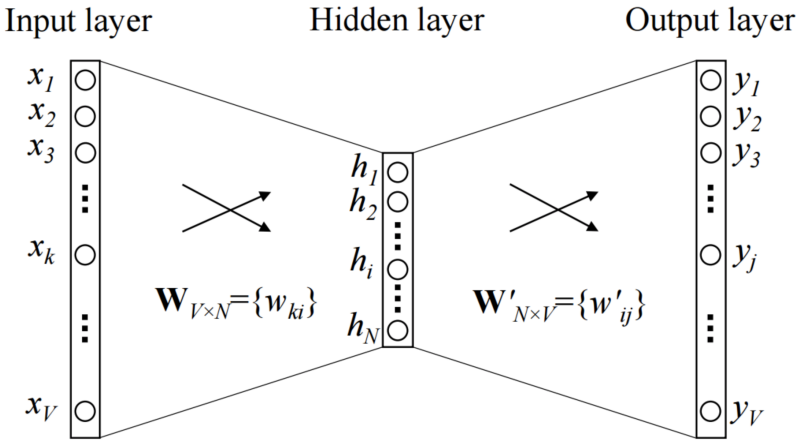

Figure 2: A CBOW model with one word in the context.

4 Models

After refining the tweets, removing links and metadata, the Skip-gram model of Word2vec was implemented on the dataset to create a model. Below is a 100 dimensional representation of the 'Melbourne'.

```
>>> model['Melbourne']
```

```
array([ 1.1955128 ,  2.156172  , -2.5918593 , -2.1451666 , -1.659118  ,
       -0.5101463 ,  1.1427345 ,  1.8097506 ,  0.7989777 ,  0.93828213,
       -0.9102811 , -0.49437764,  1.6365969 ,  0.4469444 ,  2.0494468 ,
        1.5284786 ,  1.2806247 ,  0.13305406,  1.0726079 ,  2.416711  ,
        1.5868376 ,  0.41368324,  0.73016536,  0.12340187,  2.1554594 ,
        0.33027664, -1.1811454 , -2.8899455 ,  1.0603896 ,  2.51303   ,
        0.8558165 ,  0.22636607,  2.745563  , -0.44970363,  0.20098592,
        2.153102  , -0.42428333, -0.19253883, -1.8828795 ,  0.16827443,
       -0.3510291 ,  0.43774354, -1.8374312 ,  0.2460223 ,  2.6688383 ,
       -0.2193945 ,  0.7846099 ,  1.6229521 , -0.44259644, -2.5235062 ,
        1.133652  ,  2.9692788 ,  2.60162   ,  0.8634343 ,  0.11006365,
        0.41568887,  0.2552061 , -0.52692914,  0.68827796,  0.74216765,
       -0.4878345 , -2.4649222 ,  0.38801646,  0.14129403,  0.4889239 ,
       -1.3913199 , -1.4964014 ,  2.1446564 ,  2.5550938 ,  5.2772894 ,
        0.15011099,  1.75554   , -2.9569645 , -0.86496055,  2.027168  ,
       -0.14256626, -0.0835851 , -0.52882415, -0.35895702, -0.17282951,
        1.3228668 ,  1.1968023 ,  0.8898054 , -0.5846969 ,  1.2605269 ,
        2.3877094 , -1.0491416 , -1.5062573 , -0.40664336,  1.7790266 ,
       -1.0149719 , -1.4066027 ,  0.01252248, -0.48854566, -2.6552722 ,
       -2.565324  ,  1.7439976 , -0.31473443, -0.19512999, -0.34142277])
```

Some interesting results from our model are listed below.

4.1 Similarity between words

The similarity between the two strings as parameters are the cosine similarity of their vector representations in the model, the higher the more similar.

```
>>> model.similarity('Sydney','Melbourne')
0.9460143
>>> model.similarity('woman','man')
0.8378395
>>> model.similarity('Norway','Sweden')
0.84316766
>>> model.similarity('Soccer','Basketball')
0.81141424
>>> model.similarity('Mountains','Basketball')
0.3664375
```

4.2 Similar words

```
>>> model.most_similar('money')
[('cash', 0.7468093633651733), ('funds', 0.7056224942207336), ('billions', 0.693414568901062),
('taxes', 0.6810441017150879), ('profits', 0.6670222878456116), ('wages', 0.6400871276855469),
('dollars', 0.6381756663322449), ('tax', 0.6369708776473999), ('taxpayers', 0.6341031789779663),
('credit', 0.6281141042709351)]

>>> model.most_similar('vegemite')
[('Nutella', 0.8739991188049316), ('tomato', 0.8581114411354065),
 ('butter', 0.8577016592025757), ('lemon', 0.8569393157958984), ('garlic', 0.8485199213027954),
('cheese', 0.8484674692153931), ('custard', 0.8431556224822998),
('strawberry', 0.8418306708335876), ('yogurt', 0.8385453224182129)]
```

Words such as 'Vegemite' might not have been used with the same frequency if it were not for Australian users and their tweets. Following the distribution from section 2, less common words also are used much less often. From the same batch of 4,909,917 words in 389,775 tweets only 31 of those words were 'Vegemite'. This shows how even with fairly limited dataset, the algorithm is able to make sense of the implied meaning behind words using context.

4.3 Odd one out

Least similar words outputted, some applicable primarily in the Australian context.

'food' is inconsistent with the major general outlets in Australia

```
>>> model.doesnt_match('coles woolies food aldi'.split())

'food'
```

The model also recognises cities of different countries.

```
>>> model.doesnt_match('melbourne auckland sydney brisbane'.split())

'auckland'
```

Slightly less 'Australian' animals are also distinguished.

```
>>> model.doesnt_match('koala wolf kangaroo'.split())

'wolf'
```

4.4 Calculations

As vectors representations, numerical operations can be performed on them.

The most famous one being Woman+King-Man= Queen.

```
>>> model.most_similar(positive=['woman','king'],negative=['man'],topn=5)
[('queen', 0.6177430152893066), ('prince', 0.5580594539642334), ('lion', 0.533031702041626),
('goddess', 0.5244423151016235), ('symbol', 0.5224856734275818)]
```

University - Study + Workout = Gym

```
>>> model.most_similar(positive=['university','workout'],negative=['study'],topn=5)
[('gym', 0.599767804145813), ('Pilates', 0.5258020758628845), ('training', 0.5228259563446045),
('interval', 0.5205026268959045), ('routine', 0.5178384780883789)]
```

Melbourne+NSW-Sydney = ?

```
>>> model.most_similar(positive=['melbourne','nsw'],negative=['sydney'],topn=5)
[('queensland', 0.6259955763816833), ('icu', 0.6170660257339478), ('victoria', 0.6148426532745361),
('outback', 0.6109206676483154), ('europe', 0.6066687107086182)]
```

While Victoria wasn't the first result, it shows us the limitations of using data based on short and informal sentences from Twitter users.

5 Visualisation and Result

As the words representations generated by our model were 100 dimensional, two techniques were used to reduce their dimensions while trying to preserve their differences as much as possible. We used t-SNE (t-distributed stochastic neighbour embedding) and PCA (principal component analysis).

t-SNE minimises the divergence between two distributions: a distribution that measures pairwise similarities of the input objects and a distribution that measures pairwise similarities of the corre-

sponding low-dimensional points in the embedding [4]. This is quite computationally heavy, especially considering our 100 dimensional data set as the algorithm has to compare each point to every other point multiple times. PCA takes observations into a set of linearly uncorrelated variables [5]. This in turn is greatly computationally less-expensive. We initially used PCA to reduce the 100 dimensions into 20 dimensions. Then a multicore implementation of t-SNE was applied to the resulting model.

5.1 Racist Words

Racist speech is a topic that is difficult to clearly define. It mainly depends on the context the slurs are being used. Phrases might be used by people to identify of their own race while the same phrase might be used with the intent of being derogatory. Words which identify a group directly could be racist or not racist depending on the situation. It is also important to note the difference between offensive words in general and words aimed at race specifically. Some words that are insulting are not always racist.

As such, we aim to let the model decide for itself whether some words are racist or not. A collection of predetermined racist words were set up and then located on our model.

```
>>> model.most_similar('spastic')
[('wog', 0.7162413597106934), ('fag', 0.7090408802032471), ('raccoon', 0.6972588896751404),
('foreskin', 0.6952053308486938), ('baker', 0.6854811906814575), ('anus', 0.6809008121490479),
 ('pitbull', 0.6747884750366211), ('dada', 0.67415452003479), ('stepfather', 0.6734346151351929),
 ('disfigured', 0.6713541746139526)]
```

Here 'spastic' is not a racist term but 'wog' might or might not be a similar racist pejorative.

5.2 Result

The image below on the left consists of a map of over 200,000 words plotted on a 2D graph. The image on the right represents where the racist words are present. We can see a slight clustering of these words on the top right section. But some racist words are very distant from the cluster.

There are also smaller, more distinct clusters and strips of words clumped together. This is only a representation of the words after the dimensionality reduction so information of the vectors has been lost in the process. There are also some texts present in the cloud that are not part of any vocabulary such as the strip of zeros that our text filtering process failed to detect.

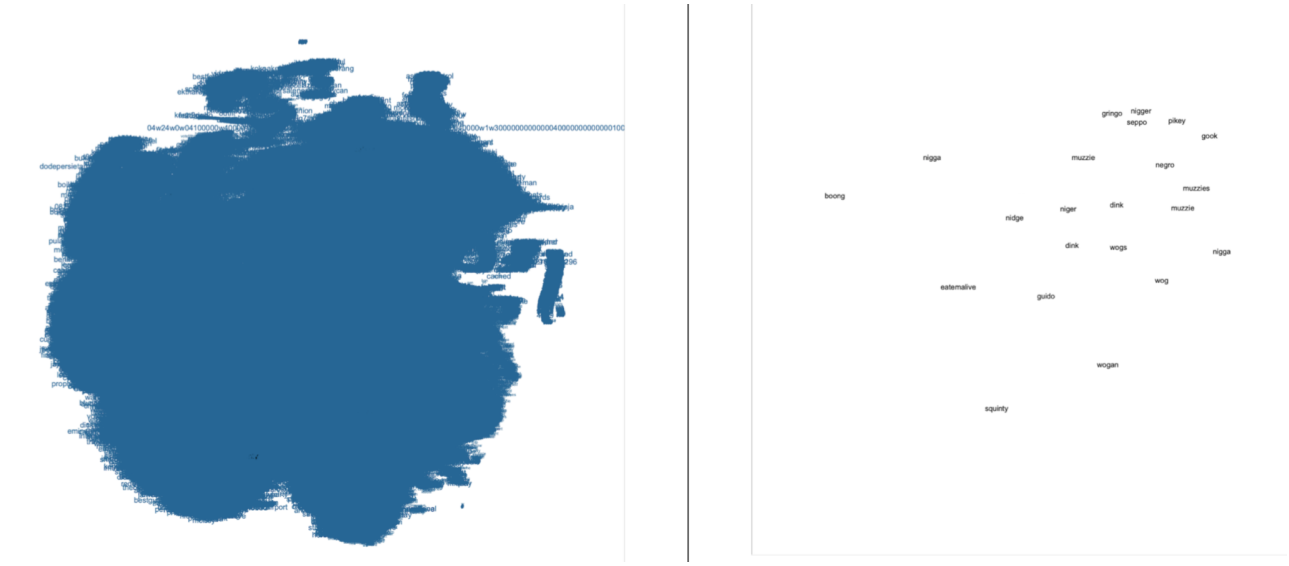

Figure 3: Side by side comparison of all words in the model vs specified racist words.

6 Future Work

A number of potential areas that can be looked into have been identified.

6.1 Emojis

While text themselves are great indicators of what opinions are being shared, there is an increasing rise in use of emojis to portray the same. A recent example of this is of Facebook implementing a set of 'reactions' for their posts. Since emojis are encoded with ASCII values, such classifications can be done. Doing this might also arise perviously unseen racist patterns which are influenced by certain types of emojis.

6.2 Trending Events

Language itself is ever changing. It is affected by recent advancements in society such as technology or even recent societal events that take place. Dictionaries are updating fairly rapidly. As such, to detect racist intent and sentiment when such events occur would be potentially worthwhile.

7 Acknowledgements

I would like to thank my supervisor Doctor Laurence Park for advising me of this opportunity and supervising me throughout this project. I would to like to thank Western Sydney University and the Australian Mathematical Sciences Institute for this opportunity.

References

[1] Levy, O. and Goldberg, Y., 2014. Dependency-based word embeddings. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers) (Vol. 2, pp. 302-308).

[2] Newman, M.E., 2005. Power laws, Pareto distributions and Zipf's law. Contemporary physics, 46(5), pp.323-351.

[3] Goldberg, Y. and Levy, O., 2014. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.

[4] Maaten, L.V.D. and Hinton, G., 2008. Visualizing data using t-SNE. Journal of machine learning research, 9(Nov), pp.2579-2605.

[5] Jolliffe, I., 2011. Principal component analysis (pp. 1094-1096). Springer Berlin Heidelberg.