University of BRISTOL

Unit Coordinator: Cameron Hall

Project Supervisor: Colin Campbell

# #tweetparsing: Twitter data mining introduction through parts-of-speech tagging, topic modelling and tweet generation techniques


| *Authors* | *Email* |
|---|---|
| V. Joshi | qk18050@bristol.ac.uk |
| C. Marsh | dg18860@bristol.ac.uk |
| J. Ebert | je16227@bristol.ac.uk |
| N. Wray | ei18063@bristol.ac.uk |

**Abstract**

The extensive growth of data on social media, in particular on Twitter has prompted extensive research into categorising tweets into parts-of-speech tags and specific topics, to better understand big datasets. Here we propose different methods for text interpretation through parts-of-speech tagging, topic models and tweet generation using 4000 of the most recent tweets of the top 20 most followed Members of Parliament in the UK on Twitter, as a database. Our approach made use of Conditional Random Fields as an introductory foray into structured output predictive models for parts-of-speech tagging which can be used to sort the data into lexical categories, and Latent Dirichlet Allocation as a framework to assign words in tweets to specific topic bands, in topic modelling. This is then furthered by showing how simple Markov chains can be used to generate tweets in the style of a particular Member of Parliament. We review our data and assess our model of lexical categorisation to be accurate to a level of 96.03%. Our studies show 70% of the top 20 Members of Parliament talk about 10 topics or more in their most recent tweets, with 28 and 2 being the upper and lower bounds respectively for the number of topics talked about in their tweets. Insightful phrases such as 'thoughts are with europe' and 'we want social distancing' may be observed through our Markov chain model of the first order in generated tweets.

# 1 Introduction

With the exponential growth of data on public opinion with the advent of the Internet in the latter part of the $20^{\text{th}}$ century, the analysis of these large datasets requires relatively unsupervised methods to reduce human errors. Parts-of-speech (PoS) tagging is a fundamental problem in natural language processing (NLP), word processors and speech-to-text programs. Twitter is a social networking site that allows its users to write short messages (*tweets*) to give opinionated views about the world and their lives, to be shared publicly or just with their *followers*, people with who the users have personal connections. With the advent of Twitter, inexpensive and rapid data collection on public views was revolutionised [O'Connor et al., 2010] with nearly 6000 tweets being tweeted across the globe every second today [Aslam, 2020]. It was Twitter's policy of not allowing more than 140 characters (later changed to 280 characters) that attracted Internet users to this form of micro-blogging as compared to traditional dispersion methods of user-generated opinions such as blogging or mailing lists. This was mainly due to the rapid and instant dispersal of tweets into the world. The techniques mentioned in this paper may be used by text interpretation businesses who require accurate PoS tagging and topic modelling for uses in word processors and speech-to-text.

We have decided to use Twitter as a data mining source as people share their opinions on a large scale from all across the world. This is because people use slang which now forms part of the daily vernacular, due to the evident ease of data collection and due to the general identification of social trends. In this paper we have focused our dataset on 4000 of the most recent tweets, collected from 20 of the most followed Members of Parliament (MPs) in the UK on Twitter. We will be using Conditional Random Fields (CRFs), a form of structured output prediction using existing NLP libraries to effect PoS tagging; creating topic modelling functions, using Latent Dirichlet Allocation (LDA) to generate the number of topics each of the 20 MPs talk about. Additionally we will create simple Markov chains to generate tweets using the database of tweets from the MPs as training data. The source code and raw data for this report may be found at: `https://github.com/vedangjoshi2000/MDM2_Proj4`.

# 2 Related Work

There exists vast research and studies on the surround themes of our project. Three key research areas were PoS tagging, topic modeling using LDA and simple Markov chains for tweet generation. Our research, as explored below, has helped us identify an area with limited recent papers. This paper aims to provide an updated look on Twitter data mining for PoS tagging.

## 2.1 PoS tagging in the literature

Twitter has been used extensively to create PoS tags not only for English tweets but for tweets in other languages too. Rehbein [2013] pioneered such work in Germany using a CRF based tagger and achieved 89% accuracy, training the data on word clusters rather than a corpus of tweets. Albogamy and Ramsay [2015] achieved 79% accuracy as compared to other studies with 49-65% tagging Arabic tweets, evaluating three different PoS taggers and perusing 390 tweets. Such PoS taggers for English tweets have

also been developed such as ARK, T-Pos and GATE TwitIE which attain 92.8%, 88.4% and 89.37% accuracy respectively [Derczynski et al., 2013]. Foster et al. [2011] used a different metric to evaluate their PoS tagger. Their use of a label attachment score (a dependency score given to a correct syntactic word in a tweet and its corresponding label (PoS tag)), gave an accuracy improvement of 4% over other previous PoS taggers evaluated by Charniak and Johnson [2005]. Although supervised PoS taggers have attained accuracy levels to 97% [Toutanova et al., 2003, Collins, 2002], the accuracy levels drop to below 97% on words within the input document/corpus in consideration and the accuracy on unknown words can be below 70% [Blitzer et al., 2006]. The advantage is that unsupervised PoS tagging is an extensively studied part of NLP [Brill, 1992, Church, 1989, Elworthy, 1994]. We decided to use tweets as an extensive source for data mining due to previous studies in data analytics, NLP and machine learning techniques [Bifet and Frank, 2010, Kumar et al., 2014, Jain and Katkar, 2015].

### 2.2 Topic modelling using LDA in the literature

There has been extensive work in the literature more notably by Mehrotra et al. [2013], investigating a new method of topic modelling using tweet pooling by hashtags which led to improved measures for topic coherence across three corpora containing tweets, without fundamentally changing the basics of an LDA model. Similar models were used on the analysis of 9 million tweets about electronic products; the authors used hashtags, mentions and emoticons present in tweets to use in opinion mining and sentiment studies [Lim and Buntine, 2014]. In our model, we clean the tweets of such extra characters; the existence of multiple hashtags in a single tweet may show the existence of hashtag-oriented spam tweets in our corpus [Sedhai and Sun, 2017]. Although LDA is a heavily investigated subject, we found no work which gave an automatic analysis of the discovered topics to discover their value. Nearly all the work in the literature used manual means to find topic titles [AlSumait et al., 2009]. The literature most commonly uses metrics of evaluation such as perplexity models and coherence values; we will be using coherence values as the sole metric in our model, as we use it to rank the number of topics corresponding to British MPs. Similar work was conducted using coherence values as a metric to evaluate tweet corpora, and the authors found the metric to be closest to manual human evaluation techniques [Fang et al., 2016].

### 2.3 Simple Markov chains for tweet generation in the literature

There have been studies using Markov models to generate sentences, the more notable one of which is BigBench [Ghazal et al., 2013] which is the current standard in large data analytics. It uses a Markov model for text generation. Subsequent work involved evaluating different methods such as Markov chains, hidden Markov chains and LDA to produce text for sentiment analysis [Maqsud, 2015]. To the best of our knowledge, the closest application of tweet generative models using Markov chains, was to create word clouds from a database of tweets, evaluating users' personal tweeting habits [Leginus et al., 2015]. We attribute this gap in the body of knowledge surrounding tweet generation to the fact that Markov chains with an order higher than 1 (chains which cannot gather information about states beyond a single preceding state), tend to replicate text from the training corpora for a particular model, which is not the aim of producing tweets.

## 3 Methodology

### 3.1 Setup

We used the Twitter API [Makice, 2009] to generate access codes to get tweets onto a Pandas dataframe using Python. Python was considered as the *de facto* programming language for our paper. Its higher level language capabilities, its easier syntax and wealth of third-party NLP libraries made it a perfect fit for our paper. Examples of tweets expressing opinions of multiple British MPs may be found in Appendix A which were used as a database for our analysis. We used British MPs as the focus of our project as MPs tend to use Twitter to promote local activities and talk about a diverse range of topics [Jackson and Lilleker, 2011]. This helps in qualitatively assessing how well our model conforms to the real-world.

### 3.2 Data Cleaning

As the tweets shown in Table 1 cannot be used in the present condition for data analysis, we used a rigorous system of filters to clean these tweets of punctuation, stray characters, removal of @ mentions, hash tags, retweets and hyperlinks. We also decided to remove strings in tweets containing numeric values as they did not add meaning to the tweet itself. An extensive filter was created to detect and remove presences of flags, emoticons, symbols and pictographs and map symbols which do not add more to the meaning of

tweets. We also parsed the tweets for stop words i.e. words like 'yeah' and 'like' which also do not contribute to the meaning of a sentence, and removed them.

## 3.3 PoS tagging

### 3.3.1 Training corpus

For training the data to implement the PoS tagging, we used the Penn Treebank corpus already included within the Natural Language Toolkit library [Marcus et al., 1993] in Python, partially because previously tagged corpora are scarce and manually tagging such large corpora takes time and effort. It was also considered as there have been many studies in PoS tagging which also use this corpus [Smith and Eisner, 2005, Goldwater and Griffiths, 2007].

### 3.3.2 Conditional Random Fields (CRFs)

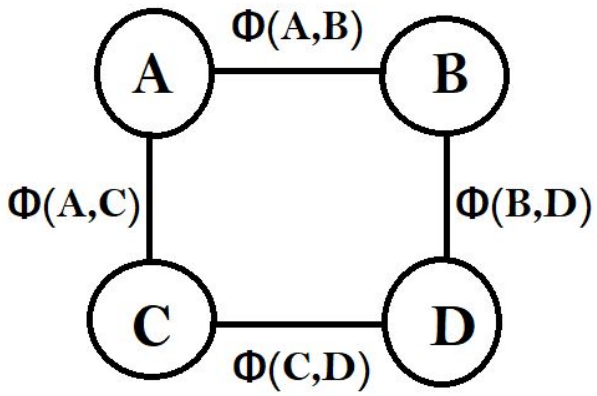


Figure 1: An MRF model with $\Phi(A, B)$ showing the weights corresponding to the edge AB, $\Phi(A, C)$ to the edge AC and so on.

Conditional Random Fields are examples of structured output prediction. Structured output prediction is defined by making a prediction based on input values ($x$) and possible labels/tags ($y$) [Nowozin et al., 2011]. This is called a discriminative model. In order to understand CRFs, we will begin by introducing the general Markov Random Field (MRF) of which CRFs are a special case. MRFs are undirected graphs, where the nodes represent random variables and the edges collectively represent dependencies between the variables [Cross and Jain, 1983]. The structure of an MRF may be observed in Figure 1 below with the nodes $A, B, C, D$ denoting random variables and the edges $\Phi(A, B), \Phi(A, C), \Phi(B, D), \Phi(C, D)$ denoting the dependencies between the nodes. The joint probability of the variables is the product of the dependency weights i.e. $\Phi(A, B)\Phi(A, C)\Phi(B, D)\Phi(C, D)$.

Let us now take an MRF and divide it into only two sets of variables, $x$ and $y$, with $x = (x_1, ..., x_n)$ and $y = (y_1, ..., y_n)$. Here $x$ is the tokenized words in the tweet and $y$ is the tag associated with the tokenized word. Let $n$ denote the number of tokenized words in the dataset. A CRF (see Figure 2) is where an MRF satisfies the property that, given the values of some $x$ in the field, the probability for any connection of $y_a$ and $y_b$ given a specific $x_j$ and $a \neq b$, is equal to the probability of a connection between $y_a$ and $y_c$, given the same $x_j$, where $y_a$ and $y_c$ are neighbours [Prasad, 2019]. These conditional probabilities are based on the weights of the edges between the $x$ values and $y$ tags, as shown in the MRF model in Figure 1.

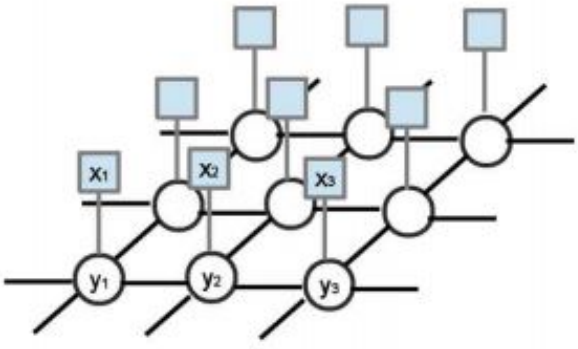


Figure 2: A CRF model with $y_1$ to $y_3$ showing the correlations between tags and $x_1$ to $x_3$ i.e. showing the words associated to the tags. Source: [Rangkuti et al., 2016]

The CRF model always models the conditional probability of the the tags associated with the tokenized tweet given the tokenized tweet, as mentioned above. A CRF looks for some sequences: an adjective tends to precede a noun, and an adverb tends to follow a verb etc. PoS taggers use certain inference algorithms to group such words (ex. adverbs and verbs & nouns and adjectives) in order, in a neighbourhood within the CRF and assign these words specific tags. This rule/grouping is represented as a 'penalty' wherein the inference algorithm gives a lower penalty to words grouped together following the above mentioned sequences and a higher penalty to words grouped together which do not follow the above mentioned sequences. PoS taggers minimize the sum of these penalties whilst tagging words [Rangkuti et al., 2016].

This explanation is to highlight the fact that PoS taggers cannot understand the meaning of a word; they cannot inherently distinguish between nouns, verbs and adjectives unless there are certain sequences in the use of vocabulary that the taggers can exploit.

Python's Packaging Index has third-party libraries to create CRF models and issue PoS tags, of which one of the more notable ones is CRFSuite [Okazaki, 2007], which we have used in this paper.

## 3.4 Topic Modelling

### 3.4.1 Real world applications

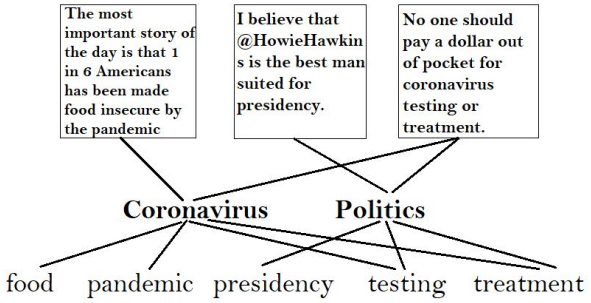


Figure 3: LDA model: The top row shows extracts from real tweets. Instead of just picking out key phrases, the model creates a hidden layer of potential topics (Coronavirus and Politics) as observed in the second row, which then breaks down into key words/phrases as shown in the third row.

With the huge amounts of data being published on the internet nowadays, it's useful to be able to categorise these into groups of similar 'topics' [Blei and Lafferty, 2009]. That is where the use of Latent Dirichlet Allocation (LDA) comes in. We discuss LDA in detail in section 2.4.3. These techniques are needed whilst recommending books to readers who want to find similar books to ones they have already read. The same concept applies to users reading the news as news providers will want to suggest similar articles that the reader might be interested in. These techniques may also be used by scientists to get recommended scientific papers corresponding to their areas of interest [Wang and Blei, 2011]. Historians also use these methods when analysing text from past years to identify events in history. Interestingly, it is also used in clustering images where an image is treated as a document [Ganegedara, 2018]. Using Twitter as a data mining tool for topic modelling is far from a novel idea, with Lau et al. [2012] using topic models to track emerging events and Twitter user trends. We attempt to mimic this approach focusing on tweets by British MPs.

### 3.4.2 Latent Dirichlet Allocation

Latent Dirichlet Allocation is a network that represents a generative model of a predetermined corpus (in our case, a corpus of tweets) made up of smaller parts (in our case, different words) [Blei et al., 2003]. A generative model is the exact opposite of a discriminative model (for an explanation, see section 3.3.2).

The 'latent' in LDA refers to a hidden layer added during the modelling process. In Figure 3, we demonstrate how the LDA model creates a hidden layer of 'topics' into which the key phrases from the corpora of test tweets are fed. These tweets are just to give an idea of how the modelling works, and were not used in our database. Figure 3 also shows a hidden layer of topics (Coronavirus and Politics) that the LDA model does not explicitly show, but human judgement and inference is required to find.

LDA uses a 'bag-of-words' representation where the order of the words in the corpus does not matter, just the word count [Yu et al., 2013]. These corpora are generally represented as word count vectors in LDA. The LDA algorithm requires only two parameters, $\alpha$ and $\beta$. Each tweet is sampled from a Dirichlet distribution with a parameter $\alpha$ while each topic is drawn from a Dirichlet distribution with a parameter $\beta$ [AlSumait et al., 2009]. In the graphical model as presented in Figure 4, the corpora of tweets is shown as $M$, each tweet consisting of $N$ words with each word in $N$ represented as $w$. Here $z$ denotes per-word topic assignment [Onan et al., 2016].

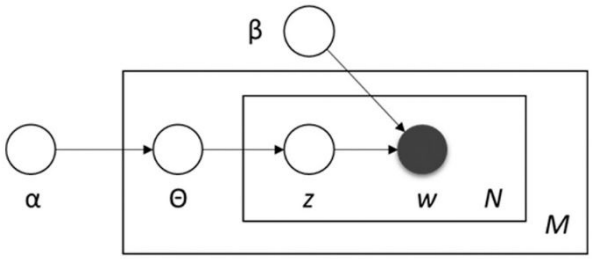


Figure 4: LDA graphical representation: The model samples $\alpha$ and $\beta$ only once and tweet-level variables such as $z$ and $w$ in every tweet and outputs conditional probabilities of topic mixtures and random Dirichlet variables. Source: [Blei et al., 2003]

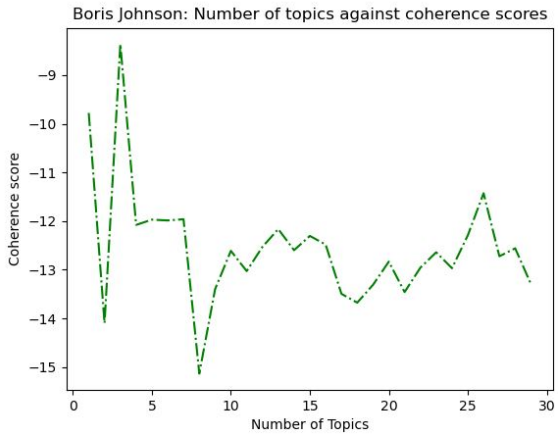


Figure 5: Coherence scores: On evaluating the semantic similarity between the words in the tweets by Boris Johnson and a range of topics, we discover the highest coherence value at the 3 topic mark.

Certain mathematical and theoretical methods of evaluation are required for topic models, due to constraints of time and effort, though none of the methods till date match the accuracy of Chang et al. [2009]'s human-in-the-loop evaluation technique. We used coherence values as a metric to evaluate each topic model produced, for each MP. Coherence value calculations assign a negative number to a range of different possible number of topics to a particular corpus. The greater that value assigned, the greater the degree of semantic similarity between high scoring words in a tweet [Kapadia, 2019]. Thus, the coherence value correlates directly to the true value of the number of topics talked about in MPs' tweets. For each MP, we evaluated coherence values for 40 potential topic numbers, and picked out the topic number corresponding to the greatest coherence value. Figure 5 shows the coherence modelling on Boris Johnson's tweets. We discovered that Boris Johnson most likely talks about three topics in his more recent tweets.

Python's third-party library, Gensim [Řehřek and Sojka, 2011] delivers powerful topic models and natural language processing models, using modern machine learning techniques. This library was used in our analyses as it was built on top of fast and memory-efficient numerical analysis libraries such as NumPy and SciPy.

### 3.5 Generated Tweets: Simple Markov chains

Markov models have been used widely to generate sentences (or in our case, tweets) which mimic certain particular styles and mannerisms (in our case, of certain twitter users) [Batool et al., 2013]. We consider in our model a corpus of tweets from 20 different MPs, each separate corpus being considered representative of the style of that MP. We construct a Markov model, of order 1. This is based off the Markov logic that the future state is completely dependent only on the preceding state in a system. We implemented the model using the following equation [Papadopoulos et al., 2014]:

$$p(s_{i+1}|s_1...s_i) = p(s_{i+1}|s_i) \forall i \in \mathbb{N} \quad (1)$$

This equation says that the probability of any future state $s_{i+1}$ depends only on its preceding state $s_i$ for all $i$ belonging to the set of natural numbers. Using the data we had already obtained from various politicians we were also able to generate potential tweets each politician may make using simple Markov chains. Taking the previous tweets from a particular politician we formed a Markov chain with each word in a tweet becoming a state and directed arrows going to the state corresponding to the next word in the sentence with a blank state to symbolise the end of a sentence. This process is then repeated for all the tweets in the data, adding to the same Markov chain. Duplicate states from a word are given an increase in probability the same as if a new state were to be added. We then give all the arrows a probability according to the equation:

$$P = \frac{1}{N \cdot D} \quad (2)$$

where $P$ is the probability of the occurrence of a particular state, $N$ is the number of potential next states and $D$ is the number of times the future state has been a duplicate. Once the Markov chain is formed we pick a starting state at random and follow the Markov chain until a blank state is found, signaling the end of the predicted tweet, outputting the states in sequence to form a sentence.

## 4 Results

### 4.1 PoS tagging

We observed that our model tokenized the words in the tweets obtained from the twitter feeds of British MPs, and cleaned the data effectively. This cleaned data was then put in a single string which was then tokenized to get the separate words to get tagged.

Figure 6 shows the tagged words in the latest few tweets by Boris Johnson. We also used CRF-

Suite's inbuilt metrics module to figure out the accuracy of the CRF model, and we found that that the model has an accuracy of 96.03% when trained on the Penn Treebank corpus. Please refer to `https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos` for the full forms of the tags used in Figure 6.

```
BorisJohnson

[('weve', 'NN'),       ('this', 'DT'),          ('the', 'DT'),
 ('got', 'VBD'),       ('morning', 'NN'),       ('nation', 'NN'),
 ('to', 'TO'),         ('i', 'NN'),             ('will', 'MD'),
 ('keep', 'VB'),       ('took', 'VBD'),         ('not', 'RB'),
 ('going', 'VBG'),     ('part', 'NN'),          ('forget', 'VB'),
 ('follow', 'IN'),     ('in', 'IN'),            ('you', 'PRP'),
 ('the', 'DT'),        ('a', 'DT'),             ('if', 'IN'),
 ('guidance', 'NN'),   ('minutes', 'NNS'),      ('you', 'PRP'),
 ('on', 'IN'),         ('silence', 'NN'),       ('can', 'MD'),
 ('social', 'JJ'),     ('to', 'TO'),            ('keep', 'VB'),
 ('distancing', 'NN') ('remember', 'VB'),       ('going', 'VBG'),
 ('and', 'CC'),        ('those', 'DT'),         ('in', 'IN'),
 ('stay', 'NN'),       ('workers', 'NNS'),      ('the', 'DT'),
 ('at', 'IN'),         ('who', 'WP'),           ('way', 'NN'),
 ('home', 'NN'),       ('have', 'VBP'),         ('that', 'IN'),
 ('to', 'TO'),         ('tragically', 'RB'),    ('you', 'PRP'),
 ('protect', 'VB'),    ('died', 'VBD'),         ('have', 'VBP'),
 ('our', 'PRP$'),      ('in', 'IN'),            ('kept', 'VBN'),
 ('nhs', 'NNS'),       ('the', 'DT'),           ('going', 'VBG'),
 ('and', 'CC'),        ('coronavirus', 'NNS'), ('so', 'RB'),
 ('save', 'VB'),       ('pandemic', 'VBP'),     ('far', 'RB'),
 ('lives', 'VBZ'),                              ('and', 'CC'),
```

Figure 6: This figure shows all the PoS tags associated with the words used in Boris Johnson's latest few tweets.

## 4.2 Topic Modelling

Our model managed to clean the data effectively and output the topics and the words in each topic band with the probability of each word being in that topic. Figure 9 shows that only 70% of the topic numbers lie above the 10 topic mark, with coherence values for all MPs ranging from -15 to -8. Jeremy Corbyn and Jeremy Hunt led the topic count, talking about 28 topics each while Ed Milliband, David Lammy and Keir Starmer talking about only 2 subjects in their most recent tweets. The abnormally low topic numbers for some MPs stems from a low $\alpha$ value which was set to the reciprocal of the number of topics for each MP separately.

```
[(0,
  '0.029*"research" + 0.028*"jenny" + 0.024*"find" +
0.022*"commitment" + '
  '0.022*"clapforourcarers" + 0.019*"prevention" +
0.018*"many" + '
  '0.018*"family" + 0.013*"duchy" + 0.011*"art"'),
 (1,
  '0.009*"remain" + 0.009*"like" + 0.009*"repeat" +
0.009*"economy" + '
  '0.009*"everybody" + 0.009*"or" + 0.009*"demand" +
0.009*"months" + '
  '0.009*"her" + 0.008*"if"'),
 (2,
  '0.064*"clapforourcarers" + 0.056*"slow" +
0.054*"home" + 0.050*"made" + '
  '0.049*"chance" + 0.037*"humanity" + 0.031*"local"
+ 0.024*"figures" + '
  '0.021*"follow" + 0.016*"complying"'),
```

Figure 7: The first 3 topic bands derived from 200 of Boris Johnson's latest tweets. The probability in front of each individual word is the probability of that word belonging in that particular topic band.

**Generated Tweets:** Such charges risk undermining efforts to pay I welcome measures needed going back to the.

**Generated Tweets:** Blessing I paid the windrush generation came together we need support for us all around.

**Generated Tweets:** Initially talked about social distancing when the labour government urgently needed for writing it with.

**Generated Tweets:** Brilliant eulogy thank you Elliot dallen for the London we want social distancing one community.

**Generated Tweets:** Be clear and thoughts are with europe or debate is strong.

**Generated Tweets:** Mike amp made an interesting and those for small businesses and corporate bailouts look like.

Figure 8: Generated tweets in the style of British MPs Jeremy Corbyn, Theresa May and Ed Milliband using simple Markov chains.

Figure 7 shows the first three topic bands containing words with the probabilities of them being in that particular band. For example, in the first topic band, there is a 2.9% likelihood that the word 'research' was in that band. Similarly there is a 5.4% chance that the word 'home' lies in the third topic band. Topic modelling does not give the names of the topics themselves but just a bunch of words with semantic similarity and probabilities as to the likelihood of their existence in a particular topic band.

### 4.3 Tweet Generation

The Markov chain text generator mostly showed nonsensical tweets, but there were a few aspects to consider as to the structuring of the tweets. We highlight certain phrases that the tweet generator came up with, which we consider to be a success given the simplicity of the model. Figure 8 shows phrases like 'risk undermining efforts', 'generation came together', 'absolutely vital resources for government', which were formed only by looking at the preceding word in the generated tweet. The more successful phrases include 'Brilliant eulogy thank you elliot dallen' and 'Initially talked about social distancing when the labour government urgently needed' form parts of sentences which may truly originate from real tweets.

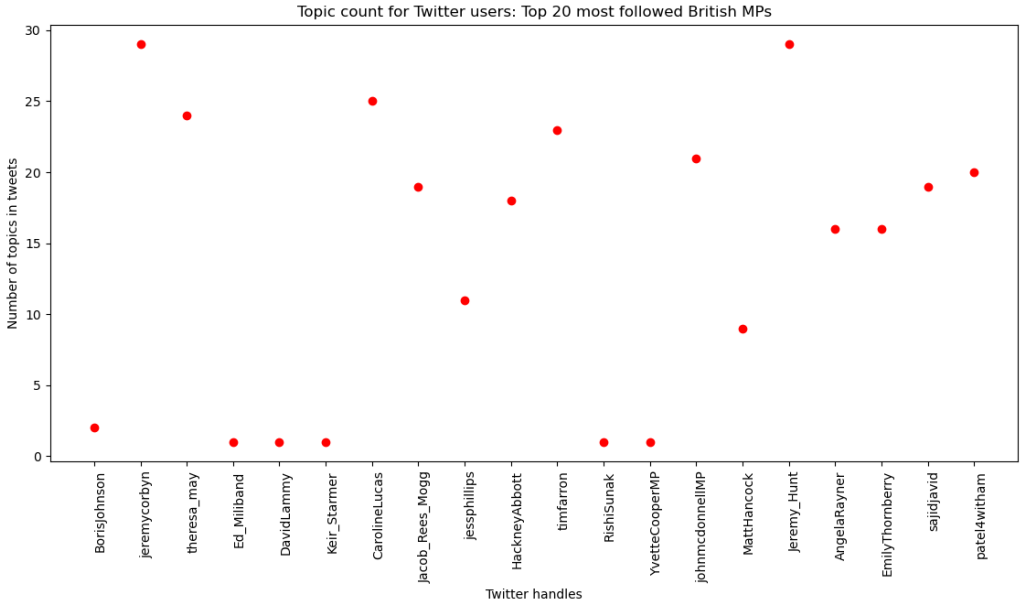


Figure 9: Topic count: The figure plots each of the 20 MPs against the number of topics determined for each MP according to the coherence evaluation. Data was taken from 4000 tweets with $\alpha = 1/\text{Number of topics for each MP}$.

## 5 Discussion

### 5.1 CRF-based PoS tagging

Our model shows the potential of CRF-based PoS tagging which can effectively tokenize and clean tweets. To further develop our findings, we could have included more tweets in our database. We also noticed that contractions like 'we've' and 'i'm', when cleaned, resulted in such words being invariably classed as nouns, which is a limitation of the model. Through collecting more data, one potential improvement would also be to use tweets from a diverse demographic of people. Using British MPs was ideal for this project, but studies could develop a better model using Twitter users from different backgrounds and in different occupations, where we can expect greater variations in literacy skills. This may introduce colloquialisms that the model cannot comprehend, but an updated PoS tagger should be able to identify these new forms of the daily vernacular.

### 5.2 LDA: Topic modelling

There are many limitations of our topic, the most pertinent of them being the fact that the number of topics in a model is fixed as remains unchanged over time. To evaluate such a model successfully, the number of topics must be known ahead of time, which in our case, was extremely hard to find. The other main limitation of LDA is that it cannot describe correlations between topics.

### 5.3 Markov chains: Tweet generation

Generating tweets using simple Markov chains showed limited real-world success, but given the simplicity of the model, our findings show the model's word prediction capability, which may be useful for text-to-speech capabilities or word processing sys-

tems. While many successful phrases can be pulled from our findings, it's observable that many of these phrases are found midway through the sentence. To improve this, future studies could prime the generator to use a selection of words to start with, which would help guide the following states in the Markov chain to form a logical sentence. Naturally more data would be required here, which was not feasible for this model. Some data showed the same word being independently generated multiple times during tokenization. Though this is a theoretical drawback of the model there have been studies with similar deficiencies such as the model described in [Klein and Manning, 2005], which attained significant improvements over prior unsupervised models. In the real world, our findings suggest similar models could accurately categorise groups of topics on the internet. This could be used on other social media networks to determine frequently discussed topics, but more importantly, to determine the stance of individual statements.

# 6 Conclusion

This paper demonstrates using mined data from Twitter to create a CRF-based PoS tagger and a basic topic model using LDA by using 4000 recent tweets from British MPs as testing data. With this data, we used the coding language Python to firstly clean data by removing all irrelevant stop words, symbols and other characters etc. We then trained the data to implement the PoS tagging for which we used the Penn Treebank corpus. The model produced tagged words for all the tweets for all MPs which when we used CRF-Suite's inbuilt metrics module to compare against tags in the training corpus, we found an accuracy of 96.03%. The second part of our study was using LDA to effect topic models. Our results showed that 70% of the MPs talked about more than 10 topics in their recent tweets with Jeremy Corbyn and Jeremy Hunt leading the topic count, talking about 28 topics in their recent tweets. Lastly, we explored tweet generation using simple Markov chains. Our model tokenized the words in the tweets and cleaned the data effectively. The tweets that were generated largely followed a normal sentence structure and parts of the tweets did make sense which was very successful considering the simplicity of the model. Phrases such as 'risk undermining efforts', 'initially talked about social distancing' and 'brilliant eulogy thank you elliot dallen', were obtained in generated tweets in the style of Ed Milliband, Theresa May, Boris Johnson etc. These well-constructed phrases may form parts of real-world tweets by British MPs, which contributes largely to the success of our Markov model.

# Bibliography

Albogamy, F. and Ramsay, A. (2015). POS tagging for Arabic tweets. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 1–8, Hissar, Bulgaria. INCOMA Ltd. Shoumen, BULGARIA.

AlSumait, L., Barbará, D., Gentle, J., and Domeniconi, C. (2009). Topic significance ranking of LDA generative models. In *Machine Learning and Knowledge Discovery in Databases*, pages 67–82. Springer Berlin Heidelberg.

Aslam, S. (2020). Twitter by the numbers: Stats, demographics & fun facts. URL `https://www.omnicoreagency.com/twitter-statistics/`.

Batool, R., Khattak, A. M., Maqbool, J., and Lee, S. (2013). Precise tweet classification and sentiment analysis. In *2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*, pages 461–466.

Bifet, A. and Frank, E. (2010). Sentiment knowledge discovery in twitter streaming data. In *Discovery Science*, pages 1–15. Springer Berlin Heidelberg.

Blei, D. M. and Lafferty, J. D. (2009). Topic models. In *Text mining*, pages 101–124. Chapman and Hall/CRC.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Blitzer, J., McDonald, R., and Pereira, F. (2006). Domain adaptation with structural correspondence learning. In *Proceedings of the 2006 conference on empirical methods in natural language processing*, pages 120–128.

Brill, E. (1992). A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155. Association for Computational Linguistics.

Chang, J., Gerrish, S., Wang, C., Boyd-Graber, J. L., and Blei, D. M. (2009). Reading tea leaves: How humans interpret topic models. In *Advances in neural information processing systems*, pages 288–296.

Charniak, E. and Johnson, M. (2005). Coarse-to-fine n-best parsing and maxent discriminative reranking. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 173–180. Association for Computational Linguistics.

Church, K. W. (1989). A stochastic parts program and noun phrase parser for unrestricted text. In *International Conference on Acoustics, Speech, and Signal Processing,*, pages 695–698. IEEE.

Collins, M. (2002). Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 1–8. Association for Computational Linguistics.

Cross, G. R. and Jain, A. K. (1983). Markov random field texture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(1):25–39.

Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 198–206.

Elworthy, D. (1994). Does baum-welch re-estimation help taggers? In *Proceedings of the fourth conference on Applied natural language processing*, pages 53–58. Association for Computational Linguistics.

Fang, A., Macdonald, C., Ounis, I., and Habel, P. (2016). Examining the coherence of the top ranked tweet topics. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval - SIGIR 16*. ACM Press.

Foster, J., Cetinoglu, O., Wagner, J., Le Roux, J., Hogan, S., Nivre, J., Hogan, D., and Van Genabith, J. (2011). # hardtoparse: Pos tagging and parsing the twitterverse. In *Workshops at the Twenty-Fifth AAAI Conference on Artificial Intelligence*.

Ganegedara, T. (2018). Intuitive guide to latent dirichlet allocation. URL `https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158`.

Ghazal, A., Rabl, T., Hu, M., Raab, F., Poess, M., Crolotte, A., and Jacobsen, H.-A. (2013). BigBench. In *Proceedings of the 2013 international conference on Management of data - SIGMOD 13*. ACM Press.

Goldwater, S. and Griffiths, T. (2007). A fully bayesian approach to unsupervised part-of-speech tagging. In *Proceedings of the 45th annual meeting of the association of computational linguistics*, pages 744–751.

Jackson, N. and Lilleker, D. (2011). Microblogging, constituency service and impression management: UK MPs and the use of twitter. *The Journal of Legislative Studies*, 17(1):86–105.

Jain, A. P. and Katkar, V. D. (2015). Sentiments analysis of twitter data using data mining. In *2015 International Conference on Information Processing (ICIP)*, pages 807–810. IEEE.

Kapadia, S. (2019). Evaluate topic models: Latent dirichlet allocation (lda). URL `https://towardsdatascience.com/evaluate-topic-model-in-python-latent-dirichlet-allocation-lda-7d57484bb5d0`.

Klein, D. and Manning, C. D. (2005). Natural language grammar induction with a generative constituent-context model. *Pattern Recognition*, 38(9):1407–1419.

Kumar, S., Morstatter, F., and Liu, H. (2014). *Twitter Data Analytics*. Springer New York.

Lau, J. H., Collier, N., and Baldwin, T. (2012). Online trend analysis with topic models:# twitter trends detection topic model online. In *Proceedings of COLING 2012*, pages 1519–1534.

Leginus, M., Zhai, C., and Dolog, P. (2015). Personalized generation of word clouds from tweets. *Journal of the Association for Information Science and Technology*, 67(5):1021–1032.

Lim, K. W. and Buntine, W. (2014). Twitter opinion topic model. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management - CIKM 14*. ACM Press.

Makice, K. (2009). *Twitter API: Up and running: Learn how to build applications with the Twitter API*. " O'Reilly Media, Inc.".

Maqsud, U. (2015). Synthetic text generation for sentiment analysis. In *Proceedings of the 6th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 156–161.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. (1993). Building a large annotated corpus of english: the penn treebank computational linguistics, vol. 19, num. 2.

Mehrotra, R., Sanner, S., Buntine, W., and Xie, L. (2013). Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR 13*. ACM Press.

Nowozin, S., Lampert, C. H., et al. (2011). Structured learning and prediction in computer vision. *Foundations and Trends® in Computer Graphics and Vision*, 6(3–4):185–365.

O'Connor, B., Balasubramanyan, R., Routledge, B. R., and Smith, N. A. (2010). From tweets to polls: Linking text sentiment to public opinion time series. In *Fourth international AAAI conference on weblogs and social media*.

Okazaki, N. (2007). Crfsuite: a fast implementation of conditional random fields (crfs).

Onan, A., Bulut, H., and Korukoglu, S. (2016). An improved ant algorithm with LDA-based representation for text document clustering. *Journal of Information Science*, 43(2):275–292.

Papadopoulos, A., Roy, P., and Pachet, F. (2014). Avoiding plagiarism in markov sequence generation. In *Twenty-Eighth AAAI Conference on Artificial Intelligence*.

Prasad, A. (2019). Conditional random fields explained. URL `https://towardsdatascience.com/conditional-random-fields-explained-e5b8256da776`.

Rangkuti, R. P., Mantau, A. J., Dewanto, V., Habibie, N., and Jatmiko, W. (2016). Structured support vector machine learning of conditional random fields. In *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, pages 548–555.

Rehbein, I. (2013). Fine-grained POS tagging of german tweets. In *Language Processing and Knowledge in the Web*, pages 162–175. Springer Berlin Heidelberg.

Řehřek, R. and Sojka, P. (2011). Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Sedhai, S. and Sun, A. (2017). An analysis of 14 million tweets on hashtag-oriented spamming∗. $Journal of the Association for Information Science and$ 1638 − −1651.

Smith, N. A. and Eisner, J. (2005). Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362. Association for Computational Linguistics.

Toutanova, K., Klein, D., Manning, C. D., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology-volume 1*, pages 173–180. Association for Computational Linguistics.

Wang, C. and Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 11*. ACM Press.

Yu, Z., Johnson, T. R., and Kavuluru, R. (2013). Phrase based topic modeling for semantic information processing in biomedicine. In *2013 12th International Conference on Machine Learning and Applications*. IEEE.

# Appendix

## A Examples of sample tweets of British MPs

| |
|---|
| **Mark Eastwood**: I would like to send my best wishes to everyone celebrating #Ramadan2020 this year. The @MuslimCouncil has published guidance to outline how Muslims can practise their faith during the holy month while keeping safe from coronavirus.? #Dewsbury https://t.co/kHwTeEVed1 https://t.co/tsJtVQFOjt |
| **Bridget Phillipson**: My latest column for @SunderlandEcho on the effects of COVID-19. The priority must be people's health, but we also need to create a path to recovery. That will only come if we protect jobs and help businesses survive the immediate challenge: https://t.co/b3TlBEDXXC |
| **Caroline Lucas**: Govt's 5 tests for lifting lockdown don't include need for locally-driven recruitment of people to trace contacts, incl environmental health officers, PHE teams & NHS volunteers Pls ask your MP to sign my EDM so further #Covid19 outbreaks are rapidly dealt with at local level? https://t.co/lmuih0q5X1 |
| **Wes Streeting**: 'Leaked Labour antisemitism report shows the community was right all along' - very powerful piece by Professor Alan Johnson. Sadly, those who need to heed its message are those least likely to read it - preferring to indulge conspiracy theories instead. https://t.co/2E2aNR68AF https://t.co/ucC2xnGNzC |

Table 1: Examples of tweets posted by some prominent British MPs expressing their opinions. [Accessed 23 April 2020 on Twitter.com]