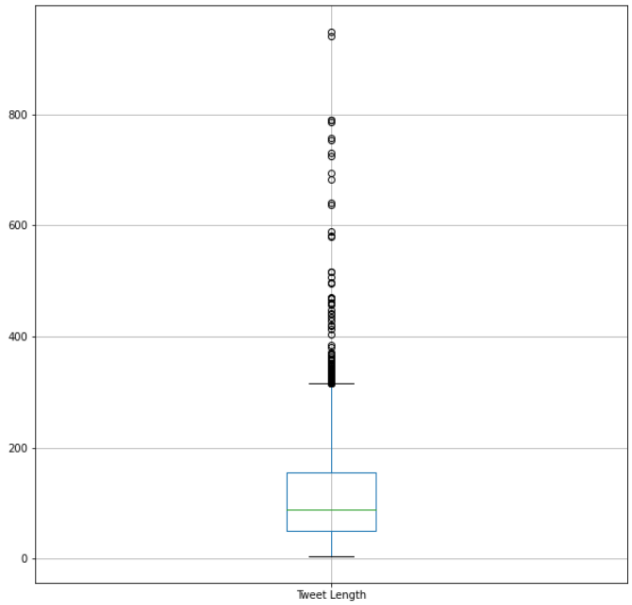
Lewis Ponsonby – 2464980P

COMPSCI4077 Web Science – Topic Modelling

Q1



The original tweet text data contained 319,469 tweets, however due to difficulties experienced in processing that volume of data I reverted to using the first 15,000 tweets from the ***NonGeoLondonJan.json*** file. The mean string length in characters across the data set was calculated as 114.837 characters, only ~41% of the maximum character limit. Although there are many outliers above the 280-character limit, I suspect this is due to the occurrence of images and links within tweets that are not included when calculating the character limit of a tweet. From the boxplot we can deduce the tweet text data has a minimum tweet length of 4 characters, a maximum of 949 characters and an interquartile range of Q1=50 to Q3=156.

Given in the current day we have huge archives of news, websites, scholarly articles, books and social media it has become increasingly difficult to discover content we are looking for. Topic modelling is the process of using a topic model to discover the topics that are hidden within a large collection of documents such as websites, books, social media etc.

The process of topic modelling is made up of three main attributes: Each **topic** is a distribution of words, each **document** is a mixture of corpus-wide topics, and each **word** is drawn from one of the defined topics. Topic structure in unstructured data is typically hidden but can be defined into a structured form using models such as Latent Dirichlet Distributions.

M x V		**M x K**		**K x V**
M = Documents	=	M = Documents	*	K = Topics
V = Words		K = Topics		V = Words

Each document can be abstracted down to a collection of individual words and therefore the grammatical context and the order of words throughout a document is not relevant for the model. The collection of words is passed through an initial pre-processing step used to prepare the data and improve model performance. Stop words that don't carry any information and occur throughout all the documents can be blacklisted, words can be

lemmatised and reduced to their most basic form e.g. running → run, speeches → speech, and punctuation or words containing special characters can be removed.

Once the data is pre-processed and tokenized, calculations can begin to be made. For each document, for each word, the following will be computed:

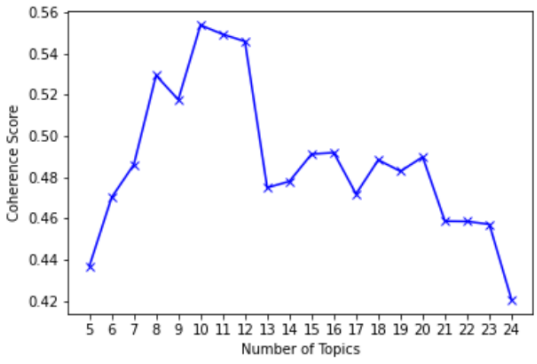
- P(Topic K | Document M)
  - The probability that a word V in document M is assigned to topic K
- P(Word V | Topic K)
  - The probability that a document M is assigned to topic K, given the document contains a given word V. If a word has a high probability of being in a topic, then other documents containing the given word are more likely to be assigned the same topic. Similarly, if a word has a very low probability of being in a topic, then it is unlikely that documents containing said word will be assigned the topic.

Using the above probabilities, the probability of word V belonging to topic K is updated:

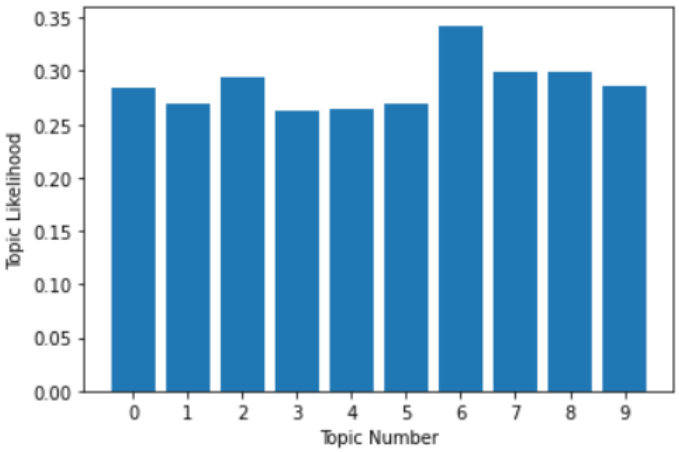
P(M belongs to K) = P(Topic K | Document M) * P(Word V | Topic K)

Larger iterations of the above calculations will allow for the LDA algorithm to converge more and provide greater results. If LDA is performed with lower iterations, topics may be prematurely calculated when further converging is possible. When iterations are exhausted, each topic can be defined as N of the most probable words to be assigned to the given topic.

Using Gensim's LDA model and coherence model I was able to select the optimal number of topics for the data set being used. I created LDA models with number of topics ranging from 5 up to 25 and calculate a coherence measure of the LDA model for each number of topics selected. From this iteration I was able to plot the coherence measures and select the optimal number of topics. From the graph I deduced 10 was the optimal number of topics



Using Gensim's LDA Model and the selected optimal number of topics I was able to use topic modelling on the tweet's text data to create the above topics. I believe the topics created are poor and would appear relatively unrelated to someone not familiar with the data set. The mean probability that a document belongs to its dominant topic was calculated as 0.288, this measure allows us to judge how confident the model was in assigning tweets topics. The graph plotted from each topics dominant topic likelihood shows that Topic 6 was the most confidently assigned, whereas all other topics were reasonably equal in performance. This is represented when the word-clouds are analysed: **dictator, atrocity** and **crime** are all dominant topics in topic 6. These words are associated together in everyday use.



By selecting all tweets assigned to topic 6, I can select examples where the topic was assigned with great success:

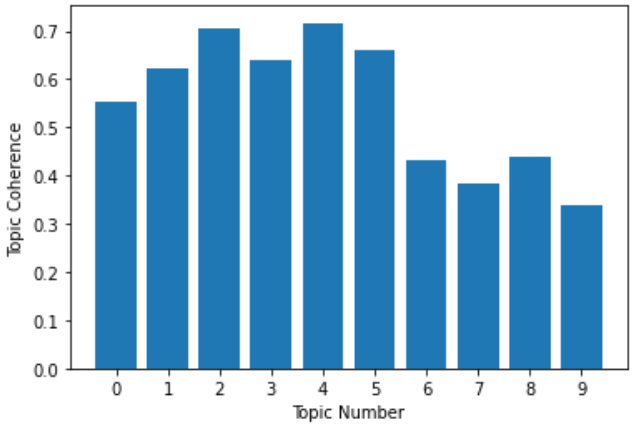
```
['champion', 'human_right', 'violation', 'dictator', 'rule', 'law', 'atrocity'
, 'crime', 'humanity', 'commits_endm', 'dictatorship', 'sijh']


['sec', 'liz_truss', 'vladimir', 'putin', 'invasion', 'result', 'quagmire', 's
oviet', 'afghanistan', 'chechnya']

['liz_truss', 'self', 'awareness', 'today', 'telegraph', 'russia', 'negotiatio
n', 'principle', 'freedom', 'democracy', 'rule', 'law', 'rule', 'law', 'card',
'liz']

['un', 'hollow', 'slogans', 'ethiopia', 'maneuvering', 'atrocity', 'genocide',
'tigrayan', 'dlm', 'sqxos']
```

The above tweets all correlate with the words: **dictator, atrocity** and **crime**. Topic 6 appears to successfully represent tweets with a theme of dystopian politics. However, if we select the worst performing topic (Topic 3), we can see counter examples of where the LDA Model was unsuccessful. The topic is made of some related words such as **PM**, **government** and **England**, likely covering the topic of British politics. However, there is also unrelated words such as **Nadal** and **United,** likely talking about tennis and football. This is likely due to the issue of performing topic modelling on data made up of short text.



The prevalence of short texts on the web has made mining the latent topic structures an important task for developing applications. However, content sparsity throughout short text and lack word co-occurrence information hinders the performance of tradition topic models. Social media posts likely lack clear context or formal structure, and this makes co-occurrence patterns hard to capture; words may overlap between posts however the lack of context makes identifying different uses difficult. The time-sensitivity or topics such as Christmas, New Years etc. creates the challenge of incorporating time stamps into certain topic model. Christmas and New Years will prove more prevalent around December, and this must be normalised across the dataset to garner relevant information. The sheer volume of short text social media data proves extremely taxing on memory requirements and data must be analysed in real-time to gain up to date context from the data.

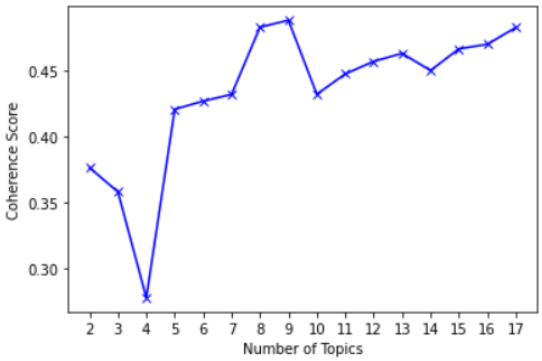
Q2

The first method I used to group short texts was to concatenate tweets together by users with more than 3 tweets in the dataset to create a new collection of documents. I restricted the selection to minimum 3 tweets to reduce the number of documents and increase the average length of document in the refactored dataset. I grouped the tweets by user as the meta-information required was available in the dataset and users on Twitter likely have a trend of tweeting about the same topics, this introduced larger documents that show more similar contexts and trends throughout.

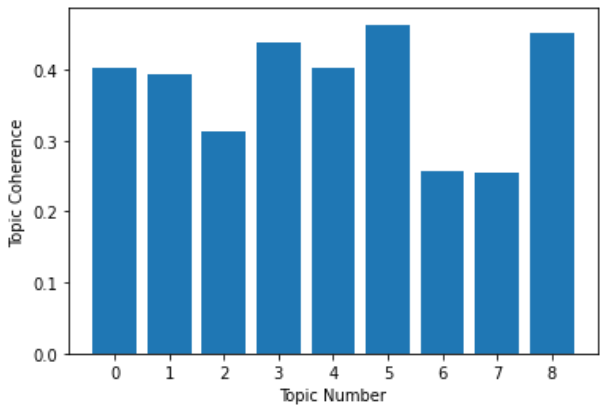
I recognised however, that restricting my grouped documents to users with at least three tweets isolates a large amount of data from users who do not meet the threshold. I decided to incorporate a Gensim Word2Vec TF-IDF model to assist in assigning the left-over tweets to the newly created documents. I trained the Word2Vec model by splitting the initial data set consisting of 319,469 tweets into a training and test split, allowing for any future cosine similarities to be made using the same context. For each tweet from users that did not meet the threshold, I calculated the Word2Vec sentence similarity with each of the refactored documents and allocated the tweet to the document that scored most similar.

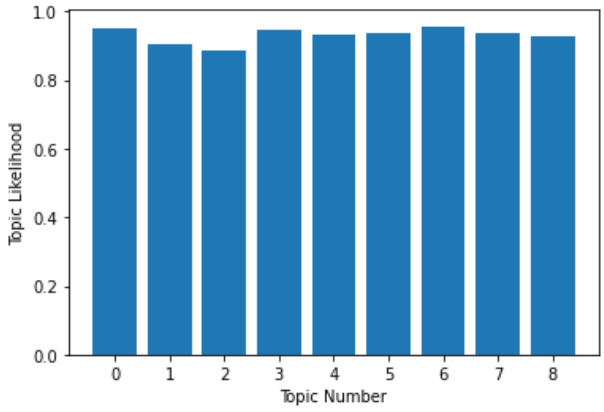
Using the above steps, I had reduced the number of documents passed into the Gensim LDA model from 15000 to 1330. The mean word count for each document with the grouped tweets, after pre-processing had been completed, had increased to 94.935.

To select the optimal number of topics for the LDA model, I used the same method as in Q1. Using Gensim's CoherenceModel a coherence measure was calculated across a range of topic numbers. From plotting the data, I ascertained that the optimal number of topics for this topic model would be 9.



Using Gensim's LDA topic model and my implementation of tweet grouping the above word clouds were produced. The majority show distinct topics, with common words appearing together; I believe that someone with no familiarity with topic modelling would be able to pick out trends and similarities in each topic. The bar chart pictured shows successful labelling for a large percentage of tweets, with a mean topic percentage contribution of 93.15%.

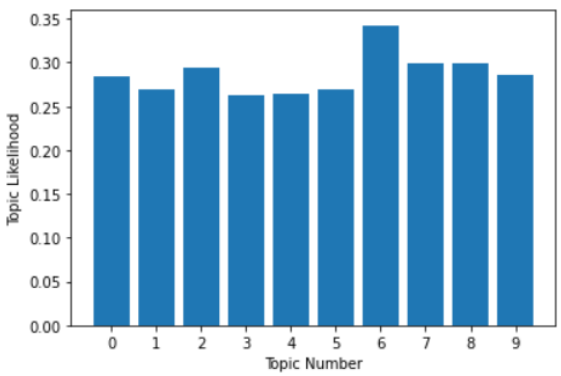




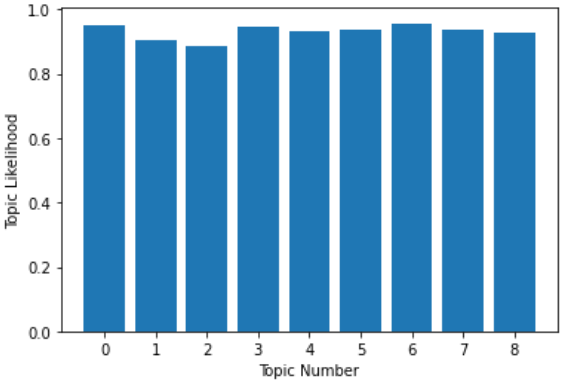
The above graphs show that topic percentage likelihood remained high throughout each topic, however topic coherence varied. Topic 6 and 7 had a low topic coherence score compared to the rest of the topics and this is represented in the lack of words that appear in the word cloud in comparison to other topics.

Q3

Non-Grouped Topic Model

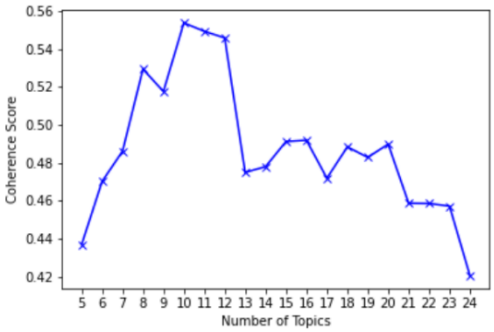


Grouped Topic Model

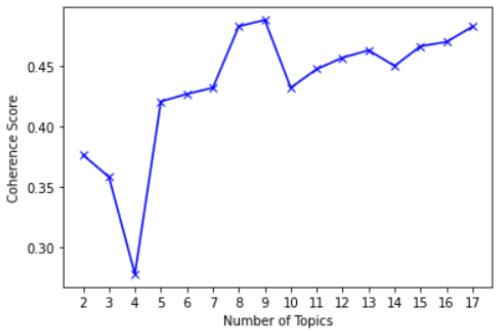


The above graphs show a large increase in Topic Percentage Contribution when using the model with grouped documents, with the mean percentage going from 28.708% without tweet grouping to 93.150% using the model with grouped tweets. This increase shows a massive increase in confidence when assigning topics for two reasons, the decrease in number of documents and the increase in size of document. The increase in the size of document allows for more cooccurrence information to be gained as more relationships between words can be formed. This increase in cooccurrence information allows for more successful linking of words when forming topics, hence the increase in topic percentage contribution. With the non-grouped model, the latent Dirichlet distribution struggled with assigning generated topics to documents consisting of such small word counts. After pre-processing and tokenization, some tweets were reduced down to only several words, likely causing issues with gaining topic information for future iterations and finally assigning a topic that fits best.

Non-Grouped Topic Model



Grouped Topic Model



The charts above show the difference in coherence score for both models. Although the non-grouped model has higher coherence scores overall, the scores are relative. The non-grouped model peaking at a higher number of topics shows the model requires more distinct topics to model keywords that are likely linked. Whereas the grouped model peaks in coherence at a lower number of topics as the increased document size increases the cooccurrence information available to the model, linking words that would previously been identified as belonging to different topics.

The non-grouped model shows some good topics and some bad topics. As discussed above, topic 6 shows several linked words such as **dictator**, **atrocity** and **crime**. However, several topics show no coherence whatsoever. Topic 1 has no distinct keywords, words such as **woman, man, guy** and **home** likely appear in tweets throughout the dataset and are words that should have been removed due to normalisation of term frequency. Topic 3 shows conflicts of sport (United and Nadal) and politics (Government, England, PM). Topic 4 contains tennis player **Medvedev** but also **football**, despite it being unlikely that tweets will appear containing tennis *and* football. Topic 7 shows several words such as **player**, **game**, **money** and **team** that could be inferred as football related as well as **week**, **month** and **year**, likely time focused. Overall, the ungrouped model shows some good topics but for the most part it is a poor example of topic modelling.

The grouped model shows several coherent topics. Topic 0 contains several keywords linking Chelsea Football Club – **cfc** and **chelsea** – as well as conservative politics – **johnsonout** and **tory**. Topic 1 is very similar to topic 6 from the ungrouped model. Topic 2 links keywords **brunch, breakfast** and **bottomlessdeal**, all correlating well with each other. Topic 8 links tennis players **Nadal** and **Medvedev,** alongside political policy related words such as **Boris**, **health, failure** and **business**. However, other topic numbers such as topics 4, 6 and 7 lack distinct content.

```
['chelsea', 'point', 'thomas', 'tuchel', 'international', 'break', 'cbd', 'vzw', 'xc', 'cfc
', 'ktbffhfrank', 'shock', 'everton', 'decision', 'gift', 'thomas', 'tuchel', 'chelsea', 'p
remier_league', 'bonus', 'lkebtteubl', 'cfc', 'ktbffhmalang', 'sarr', 'matchday', 'ritual',
'zoc', 'cfc', 'ktbffhhow', 'frank_lampard', 'tottenham', 'race', 'place', 'vmf', 'wwxv', 'c
fc', 'ktbffhwomen', 'match', 'report', 'aston', 'villa', 'chelsea', 'vgoxtrdunk', 'cfc', 'k
tbffhchelsea', 'coach', 'club', 'premier_league', 'rival', 'cfc', 'ktbffhalvaro', 'morata',
'transfer', 'warning', 'tottenham', 'antonio', 'conte', 'verdict', 'ldlkknnyaj', 'cfc', 'kt
bffhchelsea', 'statement', 'transfer', 'ncdmdf', 'jre', 'cfc', 'ktbffhthomas', 'tuchel', 'c
helsea', 'game', 'changer', 'premier_league', 'switch', 'ogallntmuf', 'cfc', 'ktbffhplaymak
er', 'van', 'der', 'vaart', 'chelsea', 'star', 'ideal', 'attacking', 'partner', 'harry', 'k
ane', 'oy', 'svgh', 'cfc', 'ktbffhousmane', 'dembele', 'chelsea', 'transfer', 'edge', 'bid'
, 'man_utd', 'stance', 'ae', 'aihixtk', 'cfc', 'ktbffhreport', 'chelsea', 'player', 'contra
ct', 'offer', 'year', 'kapuvp', 'log', 'cfc', 'ktbffhthree', 'chelsea', 'player', 'everton'
, 'transfer', 'frank_lampard', 'decision', 'svywfj', 'cfc', 'ktbffhreport', 'club', 'week',
'chelsea', 'player', 'month', 'loan', 'kb', 'ovhuu', 'cfc', 'ktbffhtottenham', 'chelsea', '
repeat', 'deal', 'trigger', 'fabio', 'paratici', 'transfer', 'rethink', 'hyyvjy', 'cfc', 'k
tbffhfabregas', 'chelsea', 'fan', 'teenager', 'fa', 'youth', 'cup', 'liverpool', 'gbhufzl',
'cfc', 'ktbffhmarina', 'granovskaia', 'surprise', 'chelsea', 'transfer', 'solution', 'probl
em', 'ek', 'hgmxji', 'cfc', 'ktbffhfa', 'youth', 'cup', 'report', 'liverpool', 'chelsea', '
ixoct', 'pmxn', 'cfc', 'ktbffhthomas', 'tuchel', 'barcelona', 'call', 'signal', 'conte', 'c
helsea', 'repeat', 'gav', 'cfc', 'january', 'transfer', 'deadline', 'mtualgs', 'cb', 'cfc',
'ktbffhhayes', 'blue', 'performance', 'cup', 'txf', 'cwgk', 'cfc', 'ktbffhantonio', 'rudige
r', 'honour', 'chelsea', 'brace', 'january', 'transfer_window', 'ldkwzvew', 'cfc', 'ktbffhf
rank', 'lampard', 'everton', 'call', 'gift', 'marina', 'granovskaia', 'chelsea', 'windfall'
, 'tuchel', 'plan', 'rhsxnyh', 'cfc', 'ktbffhthomas', 'tuchel', 'answer', 'issue', 'gjckqns
', 'xy', 'cfc', 'ktbffh']
```

The above list is an example of a document assigned to topic 0 by the topic model. Given that keywords belonging to Topic 0 include **cfc** and **chelsea**, this document shows a perfect example of tweets being successfully grouped together and assigned a topic. Although the document features 'cfc' and 'chelsea' directly, it also features many other words that were could have skewed the result but were recognised as similar words. **Tuchel**, Chelsea's manager, **Granovskaia**, Chelsea's current director, **Lampard,** an ex-Chelsea player - as well as several other keywords appear, all linking to the topic content. Although many non-Chelsea words appear, they are still related to football. The Word2Vec model will have identified these words as similar vectors and picked this document as the most alike. It appears that every single tweet grouped into this document was successfully identified and assigned as topic 0.

Overall, the grouped model showed far more coherence in the topics created, as well as more examples of tweets assigned the topics created. The concatenation of tweets into larger documents allowed for more cooccurrence relations to be formed and better topics created overall. In the topic model without tweet grouping, the texts being modelled are too short to consistently form good topics. In order to achieve better distributions, pre-processing is implemented to remove stop words, punctuation and in certain models – words that appear consistently throughout. Although this pre processing increases performance, it also reduces the size of already short text inputs, in some cases reducing to documents consisting of no tokens at all. The lack of tokens to perform calculations with means words that are related are often not identified and assigned to different topics. Topic model results would have been improved providing a larger file input was used, however topic modelling is an extremely time consuming and CPU intensive process.