Web Science

CourseWork4: Topic Modelling

StudentID: 2588922B

Student Mail: 2588922b@student.gla.ac.uk

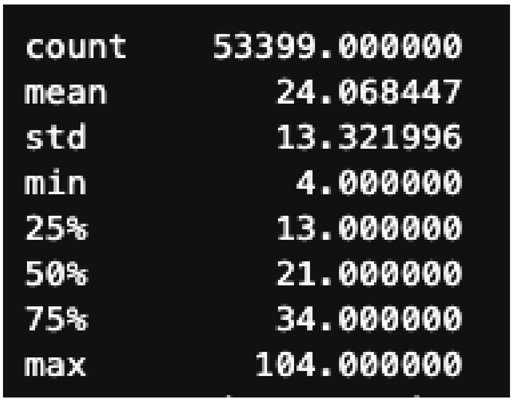
Question 1:

Topic Modelling is the process of automatically discovering the topics in a web collection of documents, which is unstructured and huge, thus, topic modelling is a form of unsupervised learning. A topic is a distribution over a fixed vocabulary. Through topic modelling we can organise the collection based on the different topics that we discovered. It does not require any prior annotation of the documents as we infer the topics after the analysis of the given texts in the collection. More precisely, we have available and work only with the text of documents and we are trying to identify the topics, the distributions of topics per document and the per-document per-word topic assignments, which are called hidden structure.

The topic modelling algorithm the we will be using in this coursework is the Latent Dirichlet Allocation (LDA). LDA is a probabilistic topic model, thus we treat our data arising from a generative process including observed (terms in documents) and hidden variables (in our case the above-mentioned hidden structure). Intuition behind LDA is that every document is composed of various topics, and also each topic has corresponding terms (weighted accordingly) that describe it. A key aspect of LDA is that every document in the collection share the same set of topics, but differently weighted.

Twitter data generally are noisy and consist of abbreviations, shortened words and many words that could be characterised as slang. Moreover, the most important issue is that tweets are short as the limit of characters is 280 per tweet, thus it’s tough to extract useful information from them, an issue that we will tackle later on with more details.

Our data set consists of approximately 54,000 tweets alongside with author of each one of them. Statistics of the data set are shown in the next image, where we can see that on average every tweet has 24 words, but also the vast majority of the tweets (75%) have less than 34 words. This word sparsity will probably lead to lack of co-occurrences of terms and will make it harder to identify the dominant topics.



```
count    53399.000000
mean        24.068447
std         13.321996
min          4.000000
25%         13.000000
50%         21.000000
75%         34.000000
max        104.000000
```

Initially we need to apply some pre-processing step in order to decrease the data size but also increase the quality of it.

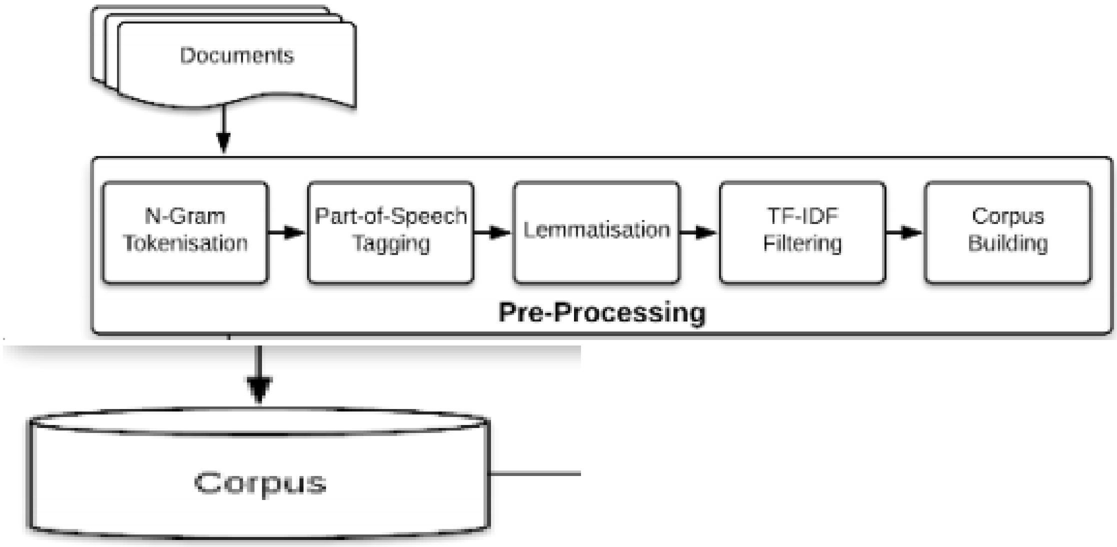
More precisely, we need to remove

- emails
- newline characters
- single quotes

After that, we need to tokenise each tweet into a list of words, alongside with punctuation and useless character removal.

Then we need to remove stop-words (apart from the ones in the given notebook, we have enhanced it with some extra words such as ‘https’, ‘co’, ‘amp’ that were observed during EDA to appear a lot of times in the data set). Create Bigram and Trigram models, two words and

three-words that co-occur frequently together respectively. We decided to increase the threshold for both bigrams and trigrams in order to decrease the probability of creating non-sense grouping. Lastly to apply lemmatization in order to match similar words to their root.
Last step of the pre-processing is to create the dictionary and the corpus we will be using.
So far, we could schematically describe our process as following



Topic Model:

After the pre-processing is over, we can move on to build and train our LDA model. We will be using the lda_model from genism library. The metrics we have decided to use in order to evaluate our models are

- Coherence, measures the level of similarity between the top scoring words in a topic, and it is closely related to human's judgement of documents. Words that belong to a topic tend to co-occur in the same documents. We use 'u_mass' option, the closer to 0 the better coherence, given by the formula

$$C(t;V^{(t)}) = \sum_{m=2}^{M}\sum_{l=1}^{m-1}\log\frac{D(v_m^{(t)},v_l^{(t)})+1}{D(v_l^{(t)})}, \qquad (1)$$

- Perplexity, measures the degree of surprise a model poses on data that it has not seen before, (entropy). Indicates the generalization performance of the model, and we want it to be as small as possible, given by the formula

$$\left(\prod_{i=1}^{n} x_i\right)^{\frac{1}{n}} = \sqrt[n]{x_1 x_2 \cdots x_n}$$

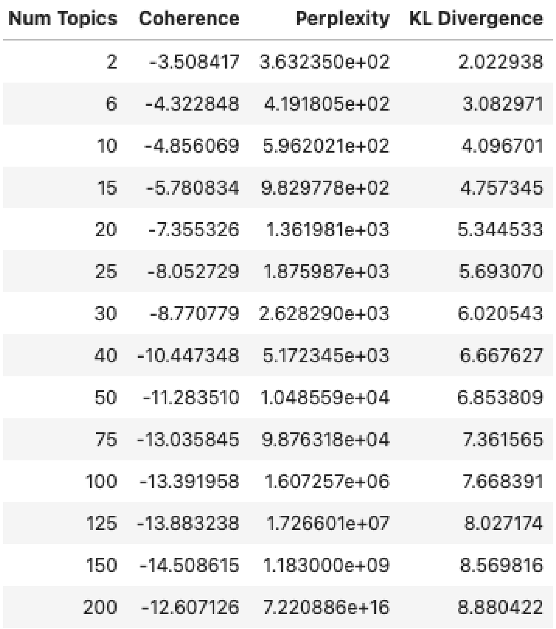
$\prod$ = geometric mean

n = number of values

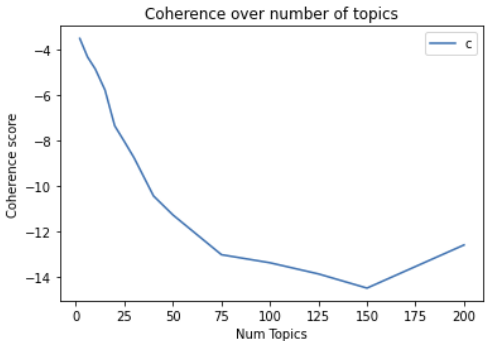
- Kullback-Leibler (KL) Divergence, measure how much a probability distribution differs from another, given by the formula.

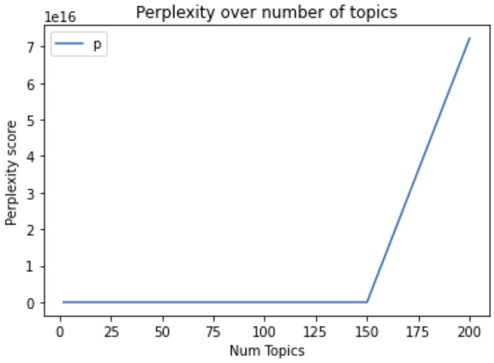
$$KL(k_i,k_j) = \sum_{w}\phi_{w|k_i}\log\left(\frac{\phi_{w|k_i}}{\phi_{w|k_j}}\right)$$

Prior to pick our model we need to optimise a crucial parameter, the number of topics, results of this process are shown in next table. Choosing the number of topics is of paramount importance, as if it is chosen wrong it might lead to more granular sub-topics We consider as optimal model the one that has coherence and perplexity closer to 0, and KL Divergence as high as possible. The above combination of metrics is achieved with 15 different topics. Which can also be observed from the following coherence and perplexity plots.



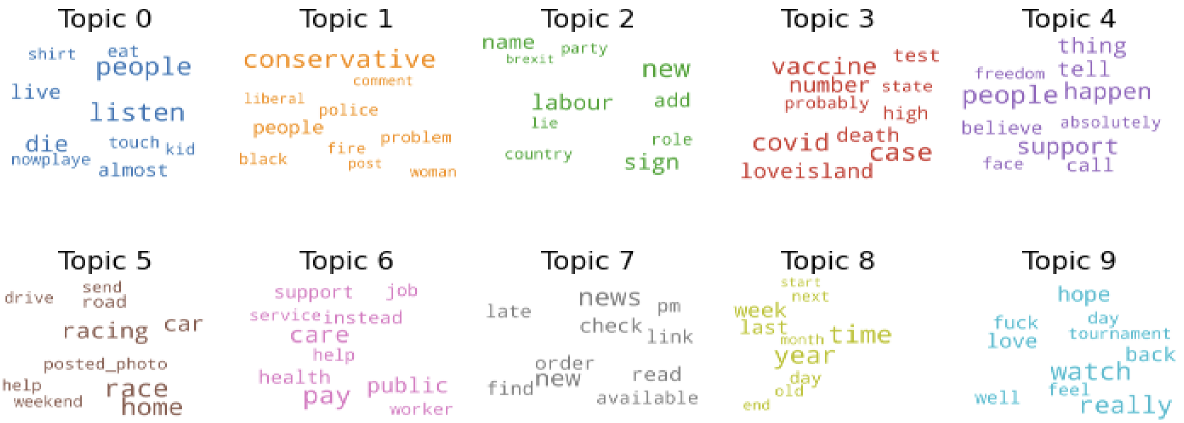
Num Topics	Coherence	Perplexity	KL Divergence
2	-3.508417	3.632350e+02	2.022938
6	-4.322848	4.191805e+02	3.082971
10	-4.856069	5.962021e+02	4.096701
15	-5.780834	9.829778e+02	4.757345
20	-7.355326	1.361981e+03	5.344533
25	-8.052729	1.875987e+03	5.693070
30	-8.770779	2.628290e+03	6.020543
40	-10.447348	5.172345e+03	6.667627
50	-11.283510	1.048559e+04	6.853809
75	-13.035845	9.876318e+04	7.361565
100	-13.391958	1.607257e+06	7.668391
125	-13.883238	1.726601e+07	8.027174
150	-14.508615	1.183000e+09	8.569816
200	-12.607126	7.220886e+16	8.880422



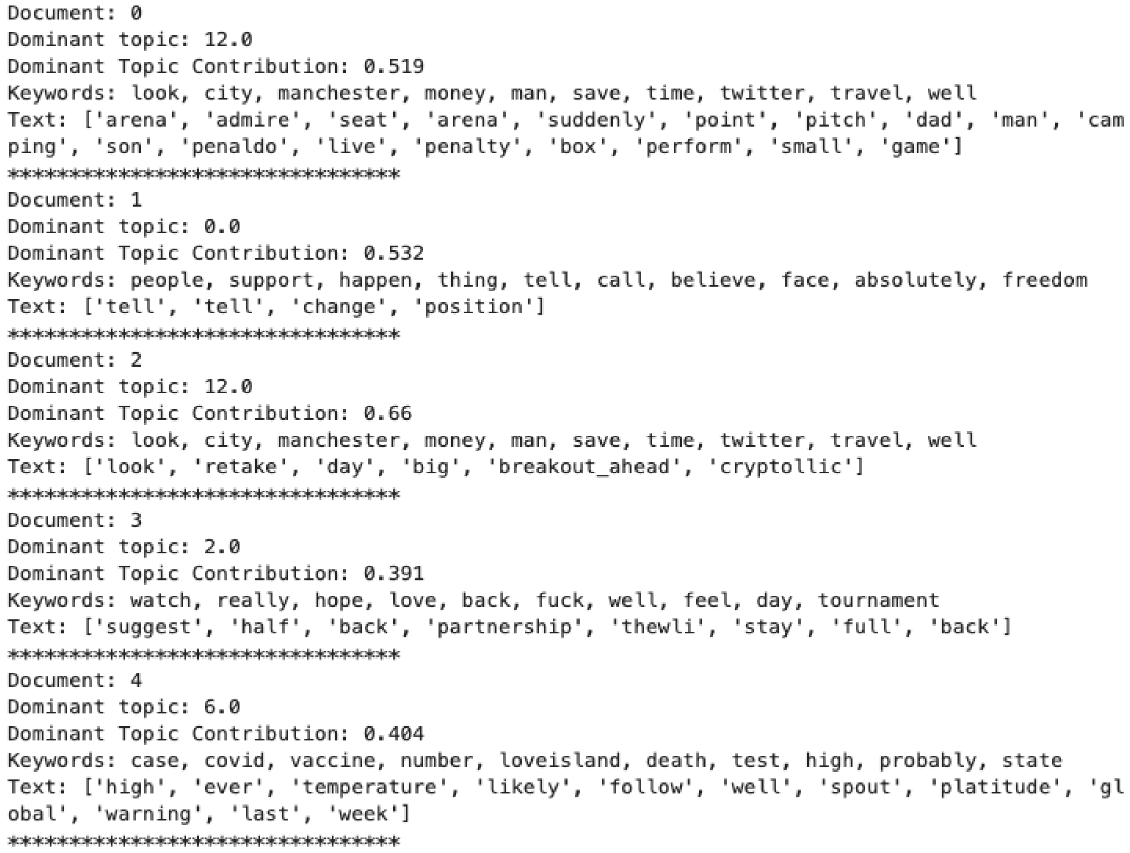


Analysis:

In first place, we can see below the keywords corresponding to the first 10 topics (the rest of them is shown in notebook) along with their corresponding weights (the bolder and larger font of a word the higher weight). Weights indicate how important a keyword is to the topic. We can observe here that it is not very obvious to identify the topic from their highest weighted keywords, as we mentioned earlier due to short document lengths. Another key observation is that we do not have many words belonging to two or more different topics, thus the number of topics we have chosen is appeared to be good enough for our given dataset.

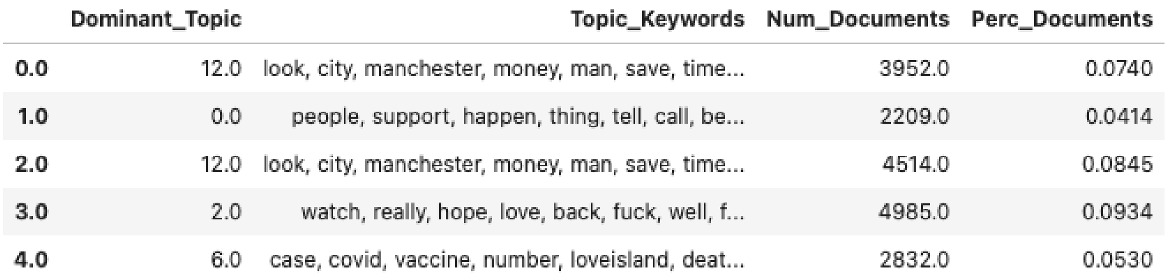


Next step is to determine what a tweet is about, more precisely what is the dominant topic in each tweet, to achieve that we will use a weighting scoring scheme for each topic in the tweet, and the topic with the highest score will be considered dominant. Below we can see the first 5 documents with their corresponding dominant topics (more documents shown in notebook) along with the percentage contribution of the dominant topic in the specific document, but also the keywords of each topic and the text of each document. We can validate here the issue of short texts, for example in Document 1 that contains (after pre-processing) only 4 terms, that the corresponding topic is something regarding UK or Economy, thus there is a clear ambiguity when the documents are shorter.



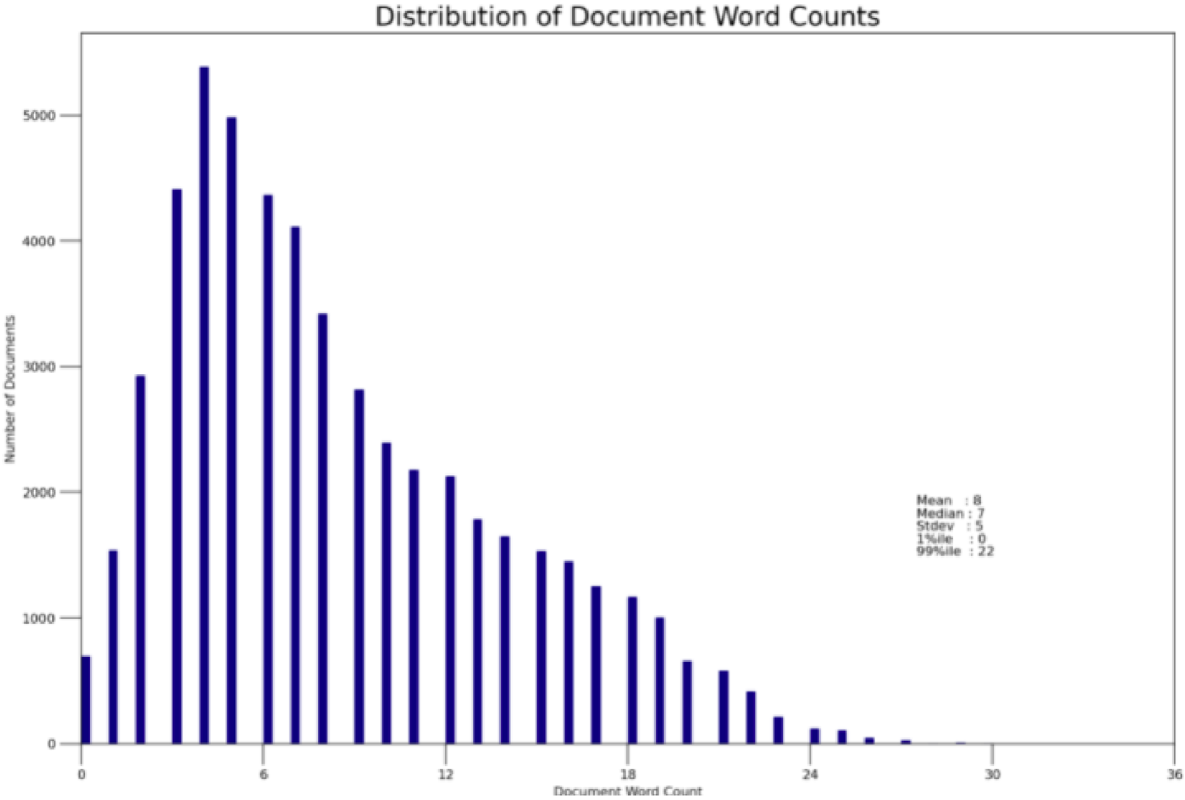
```
Document: 0
Dominant topic: 12.0
Dominant Topic Contribution: 0.519
Keywords: look, city, manchester, money, man, save, time, twitter, travel, well
Text: ['arena', 'admire', 'seat', 'arena', 'suddenly', 'point', 'pitch', 'dad', 'man', 'cam
ping', 'son', 'penaldo', 'live', 'penalty', 'box', 'perform', 'small', 'game']
********************************
Document: 1
Dominant topic: 0.0
Dominant Topic Contribution: 0.532
Keywords: people, support, happen, thing, tell, call, believe, face, absolutely, freedom
Text: ['tell', 'tell', 'change', 'position']
********************************
Document: 2
Dominant topic: 12.0
Dominant Topic Contribution: 0.66
Keywords: look, city, manchester, money, man, save, time, twitter, travel, well
Text: ['look', 'retake', 'day', 'big', 'breakout_ahead', 'cryptollic']
********************************
Document: 3
Dominant topic: 2.0
Dominant Topic Contribution: 0.391
Keywords: watch, really, hope, love, back, fuck, well, feel, day, tournament
Text: ['suggest', 'half', 'back', 'partnership', 'thewli', 'stay', 'full', 'back']
********************************
Document: 4
Dominant topic: 6.0
Dominant Topic Contribution: 0.404
Keywords: case, covid, vaccine, number, loveisland, death, test, high, probably, state
Text: ['high', 'ever', 'temperature', 'likely', 'follow', 'well', 'spout', 'platitude', 'gl
obal', 'warning', 'last', 'week']
********************************
```

In the following table, we can see the first 5 documents, the dominant along with their corresponding keywords but also the number of documents the particular topic is appeared in but also the average topic contribution it has to each document it appears.

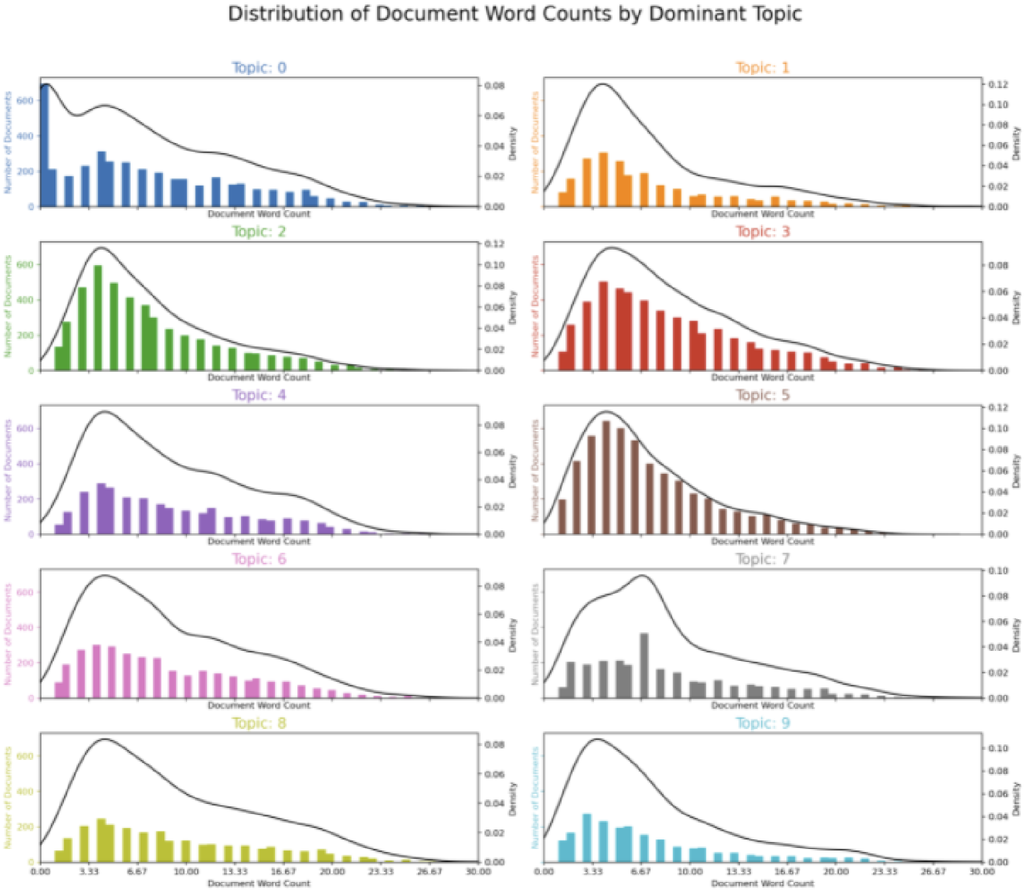


	Dominant_Topic	Topic_Keywords	Num_Documents	Perc_Documents
0.0	12.0	look, city, manchester, money, man, save, time...	3952.0	0.0740
1.0	0.0	people, support, happen, thing, tell, call, be...	2209.0	0.0414
2.0	12.0	look, city, manchester, money, man, save, time...	4514.0	0.0845
3.0	2.0	watch, really, hope, love, back, fuck, well, f...	4985.0	0.0934
4.0	6.0	case, covid, vaccine, number, loveisland, deat...	2832.0	0.0530

In the following plot we can see distribution of tweet lengths in the given dataset. It is clear, as we mentioned earlier from the statistics of the dataset that the vast majority of the tweets have less than twelve words each, while many of them have 3-6 words, which makes our task even more difficult.



As we can see in the next plots, black curve is depicting the density while the bars are showing the number of documents, for the first 10 topics given the length of the document. As we would expect, most of the topics tend to be right skewed, as majority of them have few words per document and most of the documents having few words.



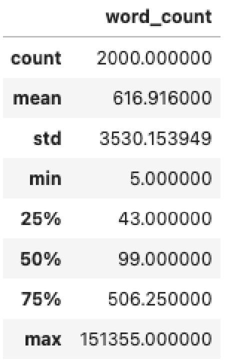
To conclude, the main issue with LDA and the given data set stems from the length of the documents (tweets) which is small and leads LDA to perform poorly, as we cannot infer the topic. That happens because of the main assumption made in LDA, which is that every text is a mixture of topics. More precisely, our documents appear to have from 3 to 6 words, so they cannot be composed of many topics and the basic assumption and procedure of LDA is failing. Thus, we need to have longer documents in order to gain better results using LDA. For this purpose, we will apply clustering in the following section and re-evaluate our model.

Question 2:

In order to enhance the performance of LDA model we need to group the tweets in order to form longer documents. In order to group the tweets, we followed the next pipeline:

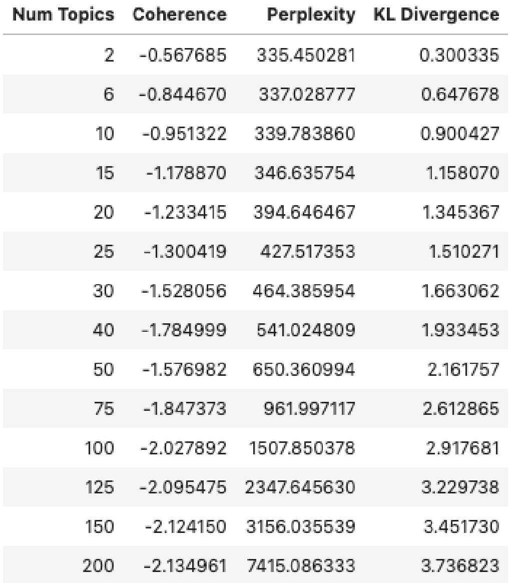
- TF-IDF vectorisation for each tweet in order to represent it as a matrix of TF-IDF features (numerical values) that we can apply mathematical operations
- Normalize the TF-IDF matrix, in order to incorporate cosine similarity as a metric, we apply the normalisation step which involves Euclidian distance (which is used in KMeans) and cosine similarity, but they have a linear relationship $\mathrm{ED}(\mathrm{u},\mathrm{v}) = 2 * \mathrm{Cos\_Sim}(\mathrm{u},\mathrm{v})$, thus we will be able to apply KMeans based on the cosine similarity

- Apply KMeans based on the number of clusters suggested in the class given instructions, which was to be as high as possible. We need to form longer documents with more dense representation but not too densely, as tweets might be connected to each other, but not that much to create even shorter documents (e.g. number of clusters 100-200). We decided to use 2000 clusters, as it was a large enough number and allowed us to run our notebook in reasonable time (compared to 10,000 clusters, which also caused memory problems)
- Joined the documents that belonged to the same cluster in order to create longer documents, statistics of the aggregated tweets are shown in the next image, in this case we have 2000 documents with 617 words on average, but also vast majority (75%) of our documents has at least 43 words. We expect the LDA performance to be enhanced, as we increased the document length significantly.

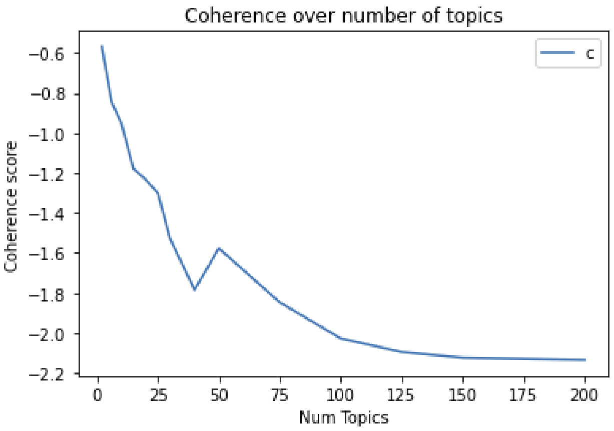


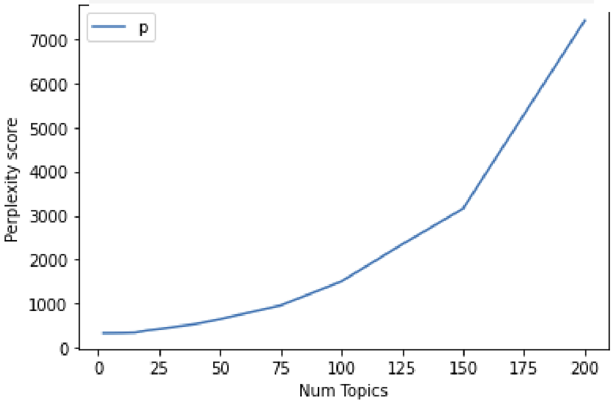
	word_count
count	2000.000000
mean	616.916000
std	3530.153949
min	5.000000
25%	43.000000
50%	99.000000
75%	506.250000
max	151355.000000

- Created the new corpus based on the aggregated tweets for each cluster
- Applied the same pre-processing we did in question 1
- Optimisation for number of topics. From the table on the right hand-side we can see that the optimal model is the one with 50 topics as coherence after that is flattening out, but also perplexity is low enough and we have a significant divergence between the various topics as we can see from KL metric. Coherence and perplexity graphs can be seen in the two figures below.

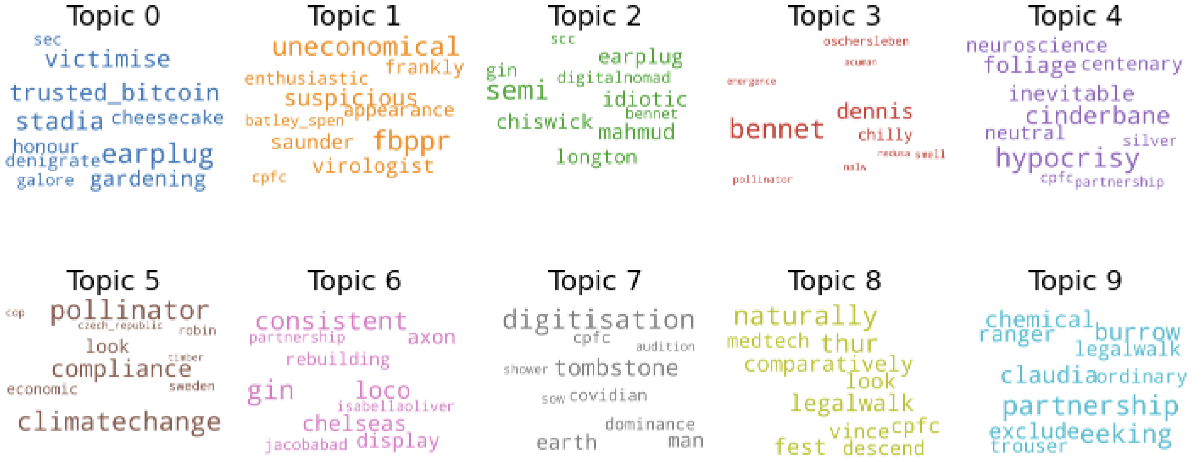


Num Topics	Coherence	Perplexity	KL Divergence
2	-0.567685	335.450281	0.300335
6	-0.844670	337.028777	0.647678
10	-0.951322	339.783860	0.900427
15	-1.178870	346.635754	1.158070
20	-1.233415	394.646467	1.345367
25	-1.300419	427.517353	1.510271
30	-1.528056	464.385954	1.663062
40	-1.784999	541.024809	1.933453
50	-1.576982	650.360994	2.161757
75	-1.847373	961.997117	2.612865
100	-2.027892	1507.850378	2.917681
125	-2.095475	2347.645630	3.229738
150	-2.124150	3156.035539	3.451730
200	-2.134961	7415.086333	3.736823

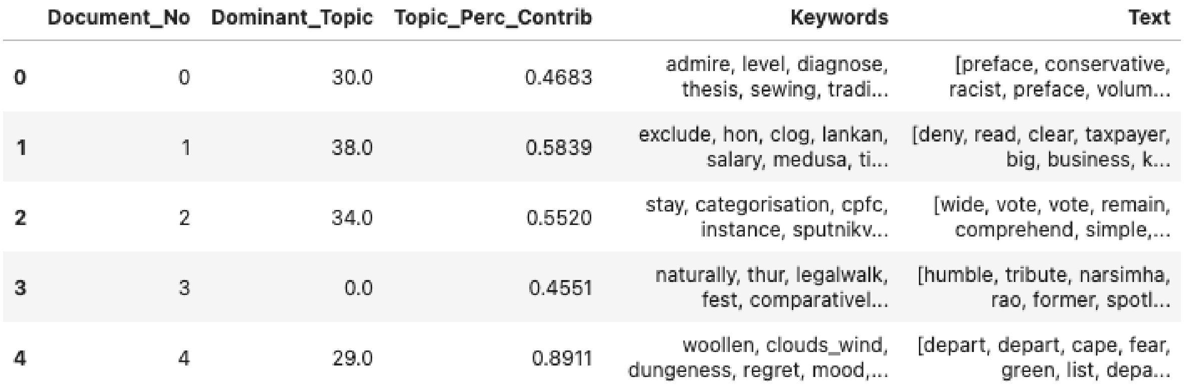




Below we can see the first 10 topics along with their corresponding terms, it is clear that we can identify and name the topics much easier than the simple LDA. Again, the size and the boldness of the fond is showing the term weight. An issue that becomes apparent here is that there some terms that make no sense (e.g. ‘fbbpr’ in Topic 1, ‘cpfc’ in Topic 8), to tackle this issue we should have removed terms with low term frequency (e.g. tf<3). However, due to pressure of time and the moment of this finding it was not feasible to re-start the process.

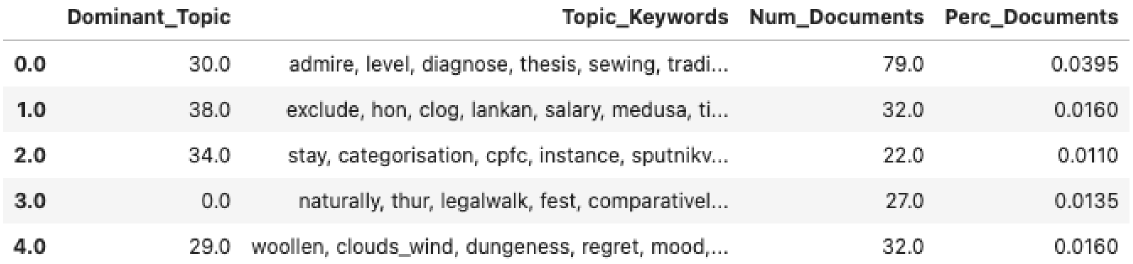


The next table is showing the first five documents along with their dominant topic, the contribution of dominant topic into the document, the corresponding keywords of the topic but also the text of the document. It is clear that the contribution of the dominant topic is significantly increased.



	Document_No	Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
0	0	30.0	0.4683	admire, level, diagnose, thesis, sewing, tradi...	[preface, conservative, racist, preface, volum...
1	1	38.0	0.5839	exclude, hon, clog, lankan, salary, medusa, ti...	[deny, read, clear, taxpayer, big, business, k...
2	2	34.0	0.5520	stay, categorisation, cpfc, instance, sputnikv...	[wide, vote, vote, remain, comprehend, simple,...
3	3	0.0	0.4551	naturally, thur, legalwalk, fest, comparativel...	[humble, tribute, narsimha, rao, former, spotl...
4	4	29.0	0.8911	woollen, clouds_wind, dungeness, regret, mood,...	[depart, depart, cape, fear, green, list, depa...

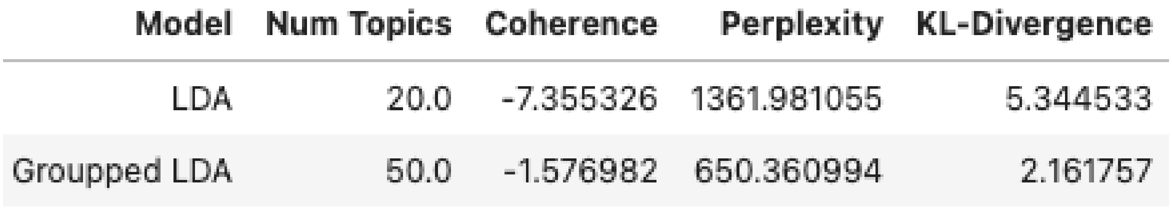
Lastly, we can see the first five documents along with their dominant topics but also the number of documents these topics appear in. It is noticeable, that the number of documents for each topic has significantly decreased, as we would expect due to the increase in number of topics, compared to LDA.



	Dominant_Topic	Topic_Keywords	Num_Documents	Perc_Documents
0.0	30.0	admire, level, diagnose, thesis, sewing, tradi...	79.0	0.0395
1.0	38.0	exclude, hon, clog, lankan, salary, medusa, ti...	32.0	0.0160
2.0	34.0	stay, categorisation, cpfc, instance, sputnikv...	22.0	0.0110
3.0	0.0	naturally, thur, legalwalk, fest, comparativel...	27.0	0.0135
4.0	29.0	woollen, clouds_wind, dungeness, regret, mood,...	32.0	0.0160

Question 3:

The next table summarizes the comparison between the two models, LDA and Grouped-LDA (GLDA), which we identified as the best options in Question 1 & 2.



Model	Num Topics	Coherence	Perplexity	KL-Divergence
LDA	20.0	-7.355326	1361.981055	5.344533
Groupped LDA	50.0	-1.576982	650.360994	2.161757

It is clear that GLDA model outperforms simple LDA, as both Coherence and Perplexity are getting significantly better. KL-Divergence poses a drop, but this is something we would expect as the number of topics is more than double in the second case, so it is normal to less different one from another.

To conclude, twitter posts are short documents thus the simple LDA model is underperforming due to its nature, the basic assumption that each topic is conformed of many topics, but with the given length of each tweet this is infeasible. The GLDA approach, is tackling this issue effectively, as the grouped documents have significantly greater length; also, they constitute of topics closely related one to another (as we used cosine similarity to group them).