Leibniz Institute for the Social Science

Faculty 3: Mathematics & Natural Sciences

Institute for Web Science and Technologies

Information/News exposure & issue salience

Master's Thesis

in partial fulfillment of the requirements for
the degree of Master of Science (M.Sc.)
in Mathematical Modeling, Simulation And Optimization

Submitted by Ziyad Meftah

First supervisor: Prof. Dr. Claudia Wagner
Institute for Web Science and Technologies

Second supervisor: Dr. Juhi Kulshrestha
Institute for Web Science and Technologies

Koblenz, January 2021

Statement

I hereby certify that this thesis has been composed by me and is based on my own work, that I did not use any further resources than specified – in particular no references unmentioned in the reference section – and that I did not submit this thesis to another examination before. The paper submission is identical to the submitted electronic version.

Koblenz, 25/02/2021

.....

(Place, Date) (Signature)

Acknowledgments

First of all, I would like to thank my thesis supervisor Dr. Claudia Wagner and my scientific supervisor Dr. Juhi Kulshrestha for their availability, advice, and involvement in this thesis, knowing the current circumstances.

I would also like to thank Dr. Sebastian Stier, Dr. Fabian Flöck, and Ina Böckmann for providing us with the necessary data and helping understand every part of it.

My thanks also go to everyone that participated in the human judgment task presented later in this thesis.

Finally, I would like to express my gratitude to my family, especially my parents and my friends, for their encouragement and support.

Abstract

In this thesis, we are interested in understanding how people's online news reading activity is related to what they believe to be the most important global issue.

To achieve that, we were provided with two datasets containing textual articles that a set of users viewed. We give details of the corpus-specific data wrangling techniques that were applied to delete any records that are not actual articles. We then describe the LDA (Latent Dirichlet Allocation) training and our combined evaluation approach that were conducted to extract latent topics in the corpus.

Lastly, we present a method that aggregates topics of the read articles to different users in the form of a matrix and then compares it with their preferred topics.

Abstract in German

In dieser Arbeit interessieren wir uns dafür, zu verstehen, wie die Online-Nachrichtenleseaktivität von Menschen mit dem zusammenhängt, was sie als das für sie wichtigste globale Thema betrachten.

Um dies zu angehen zu können, wurden uns zwei Datensätze in Form von Textartikeln zur Verfügung gestellt, die von einer Gruppe von Benutzern gelesen wurden. In dieser Arbeit geben wir nun Details zu den korpusspezifischen Datenbereinigungstechniken, welche angewandt wurden, um alle Datensätze zu löschen, bei denen es sich um keine tatsächlichen Artikel handelt. Anschließend wird das LDA-Training (Latent Dirichlet Allocation) sowie der kombinierte Evaluierungsansatz beschrieben, welche beide Male durchgeführt wurden, um latente Themen im Korpus zu extrahieren.

Schließlich stellen wir eine Methode vor, die die Themen der gelesenen Artikel für verschiedene Benutzer in Form einer Matrix aggregiert und dann mit deren bevorzugten Themen vergleicht.

Table of Contents

List of tables

List of figures

General Introduction

As we all know, technology and the way we communicate have evolved tremendously in recent decades. We have a whole range of digital devices that allow us to consult the media and access a wealth of information. However, the way we see and consume this information differs from an individual to another based on different aspects (age, gender, occupation, location ,and most importantly, preferences).

In this project, we have an opportunity for direct observation of people's behavior in an online environment. We focus particularly on news articles from a selected set of sources and how people consume them with respect to their preferences. Do people really stick to their preferences and read articles closely related to the socio-political issues they claim to be most important?

This thesis examines the following question : "***How is the issue that a person considers most important related to what they browse***?"

This is a question to ask since, despite the leading and growing role of the Internet, very little research has been done to better understand and assess its influence on its users. The attempt is to understand the data collected on users' online activity from a perspective of behavioral psychology.

To conduct our research, we consider the problem of modeling text corpora. The goal is to find some correlation between the topics generated by an unsupervised model and the critical issues to each participant taken from the conducted surveys.

Thesis structure

First, a state of the art is presented in **chapter 1** in the representation of text documents in topic spaces.

Chapter 2 describes the textual data and the methodology we used to prepare it for further analysis. It combines data wrangling in which we get rid of irrelevant articles and data pre-processing that uses natural language processing techniques to provide a clean dataset to input in our model. It also covers the model training approach and the proposed evaluation method that combines different intrinsic metrics and a human judgment technique to come up with the best models to further our analysis.

Chapter 3 presents the results of the previous chapter as labeled topics under certain classes and how they are attributed to each user in the form of user profile vectors that we will compare with their personal opinions about the most important global issues. A discussion at the end of this chapter is conducted to make sense of the results.

1 State of the art and Related Work

1.1 Topic modeling

In this work, we will focus on ways to automatically identify the topic of an article and understand its content semantically regardless of the news site structure.

To solve this problem, Data Scientists have introduced *Topic Model* algorithms: it involves modeling articles (or more generally, text documents) to determine the abstract subject of documents, using probabilistic means. Most Topic model algorithms are unsupervised, which means that there is no need to define classes before training. The algorithm automatically groups together documents that share common characteristics into several categories. In our framework, it is a matter of grouping together documents addressing the same topics without prior knowledge of them.

For this matter, we opted for LDA model that requires the number of topics to be fixed in advance. We can therefore speak of semi-supervised classification in the sense that the number of classes is given, but the topics of these classes are unknown.

1.2 Latent Dirichlet Allocation

Topic modeling (David M. Blei, Andrew Y. Ng, & Michael I. Jordan, 2003) is a probabilistic model making it possible to identify a set of themes (or topics) characterizing a set of words from a corpus of documents. The intuition is then to discover the most probable topics (with respect to the documents), grouping together the most probable words.

One possible implementation of topic modeling is LDA (David M. Beli , 2012) In order to associate words with topics, which will be assumed to be discovered according to a probability of occurrence, LDA ensures that each document is characterized by a set of topics and that each word is part of one or more topics. More precisely, LDA is a probabilistic generative model with latent variables. The parameters of this model are the number k of topics to extract, as well as two distribution parameters, α and β. The α parameter acts on the distribution of documents between the topics and the β parameter on the distribution of words between the topics.

The graphic representation of the probabilistic model LDA is given in Figure 1 (in flat notation).

The idea of this model is summarized as follows:

We suppose, beforehand, a selection of k topic distribution laws. For each document $m \epsilon M$, choose a distribution law θ of m among the topics. Then, for each word $w \epsilon W$ of m, choose a topic $z \epsilon Z$ respecting the corresponding law θ.

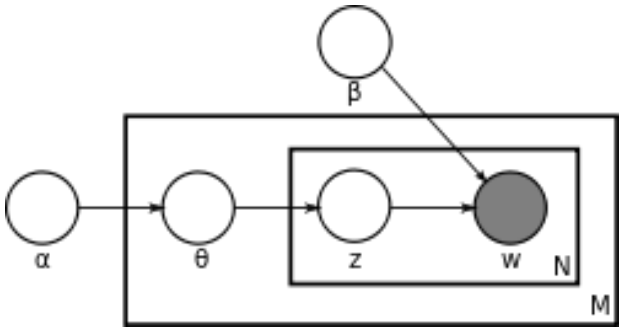


Figure 1 Graphical model of LDA in flat notation (taken from (David M. Blei, Andrew Y. Ng, & Michael I. Jordan, 2003))

1.3 Topic model evaluation

Unfortunately, the interpretation of generated topics in various applications is still not up to the mark since there is no simple or straightforward way to evaluate topic model's results comparable to human interpretability. It is because, most of the time,

the desired result is unknown and not clearly defined. Additionally, the human interpretation of results differs between people, fields, and use cases.

That is why our evaluation task is to quantify what the quality of a model means. To achieve that, we use a combination of three intrinsic evaluation metrics.

Perplexity score

It is the most widely used model evaluation metric. It consists of evaluating the probability of regenerating documents after training the model. This measure was used by the author of the LDA as a means of comparing generative algorithms with each other, considering that the more an algorithm has assimilated the internal characteristics of texts, the higher the probability of generating the corpus. The original formula used by the author of the LDA (Blei et al., 2003) is as follows:

$$Perplexity(Corpus) = e^{\frac{-\sum_{d=1}^{M} \log(p(w_d))}{N}} \quad (1)$$

Where

$$p(w_d) = \sum_{z} p(z|d) \cdot p\ (w|z) \quad (2)$$

refers to the probability of each word appearing in the test set, specifically to the LDA.

- $p\ (w|z)$ probabilistic weight of the term w in the determination of the topic z $(\sum_{w \in V} p\ (w|z) = 1)$
- $p\ (z \mid d)$ probabilistic weight of topic z for document d. $(\sum_{\in V} p\ (w|z) = 1)$

  The N of the denominator is all the words that appear in the test set reordering. It is important to note that perplexity captures how surprised a model is at new data. A model with a lower perplexity is considered a good model. However, perplexity may not be the best metric to assess topic patterns (alone) since it ignores context and semantic associations between words.

Topic coherence score

It is another metric that is more suitable for the evaluation of a model which measures the degree of semantic similarity between the words of each topic. It should make it possible to distinguish the topics which have a semantic interpretation from those which emerge from the statistical tool. Three measures demonstrated a positive correlation with human judgements based on NPMI (Normalized Pointwise Mutual Information).

- **Umass score** defined as follows:

$$U_{Mass}(W) = \sum_{w_i,w_j \epsilon W} \log \frac{D(w_i,w_j)+1}{D(w_i)} \quad (3)$$

  Where $D(x,y)$ number of documents (of the model) containing x and y and $D(x)$ number of documents (of the model) containing x.

- **UCI Score** defined as follows:

$$UCI(W) = \sum_{w_i,w_j \epsilon W} \log \frac{p(w_i,w_j)+1}{p(w_i)\cdot p(w_j)} \quad (4)$$

  Where $p(w_i,w_j)$ is the frequency of co-occurrence of terms w_i and w_j in a corpus and $p(w_i)$is the frequency of term x in the same corpus.

- $\boldsymbol{C_V}$ **Score**: a new score introduced by (Michael Röder, Andreas Both, & Alexander Hinneburg, 2015) that combines existing methods in the space of coherence measures. In their analysis, C_V achieves the highest correlation with all available human topic ranking data. It uses sliding windows to create virtual documents based on the window size.

  For example, a sliding window size of 10 words would move along words that make a topic and create vectors consisting of 10 words. These vectors are then

compared with one another with cosine similarity and then averaged into one single C_V score.

Word intrusion

"This technique measures how semantically 'cohesive' the topics inferred by a model are and tests whether topics correspond to natural groupings for humans." (Identification of Topics and Their Evolution in Management Science, 2018)

For that, we take one keyword with high probability from a topic and place it within other keywords in another topic in which it has a low probability.
The model precision is calculated as such :

$$Model\ precision = \frac{correct\ guesses\ of\ true\ intruder}{all\ possible\ guesses}$$

We ask humans that have no idea what the topics are to find the intruder, and we assign 1 when they give right answer and 0 when they don't. The higher the score the better the model.

1.4 Related work

A lot of work has already been done in identifying predefined socio-political issues in news articles.
The most occurring idea was to use Boolean search which is nothing but looking whether a word is present or absent in a document. For that, (Neuman, Guggenheim, Jang, & Bae, 2014) created, for each issue, a set of key identifying terms or phrases unique to that issue but didn't specify how these words were identified.
In the political setting, (Xinxin Yang, Bo-Chiuan Chen, Mrinmoy Maity, & Emilio Ferrara, 2016) used the same technique by manually looking for the most frequent words that could be indicative of specific topics and sound meaningful to ordinary readers. Similar to that, (Yeojin Kim, Chris J. Vargo, William J Gonzenbach, &

Youngju Kim, 2016) built their issue's keywords based on lexicon-based lists from previous agenda-setting research. Another approach was introduced by (Haewoon Kwak, Jisun An, Joni Salminen, Soon-Gyo Jung, & Bernard J. Jansen, 2018) where they built a set of 11 common topics and collected co-mentions using word embeddings (i.e., topics that are mentioned together with any of the daily top 100 popular topics per country). Then, for each co-mention, they computed the distance from each topic and assigned the topic to the shortest distance.

Finally, and in a semi-supervised setting, (Ivan P. Yamshchikov & Sharwin Rezagholi, 2018) trained a set of 7 binary text classifiers based on convolutional neural networks on annotated sentences and applied these classifiers to the non-annotated sentences. Also, (Aritz Bilbao-jayo & Aitor Almeida, 2018) did the same thing by using convolutional neural networks with word embedding for discourse classification where sentences are taken as inputs. Their model was trained using election manifestos annotated manually by the Regional Manifestos project.

Taking into consideration all the above, what makes our methodology stand out from the previous works is that it focuses on full-length articles as opposed to most of the previously mentioned papers that focus only on sentences such as tweets. In addition, it doesn't involve training our model on annotated documents and can be used at any time without having to update the keywords of any topic.

2 Methodology

This section describes the architecture of our approach and the learning models we have developed. This approach responds to the problem of identifying topics in three different parts, namely:

(1) What are the tools and resources necessary to wrangle and pre-process our data and reduce the vocabulary used in the corpus?

(2) What is the unsupervised learning model to be used to learn the topics discussed in the corpus?

(3) How to evaluate the performance and quality of our models' results?

Graph 2 below displays an overview of our procedure.

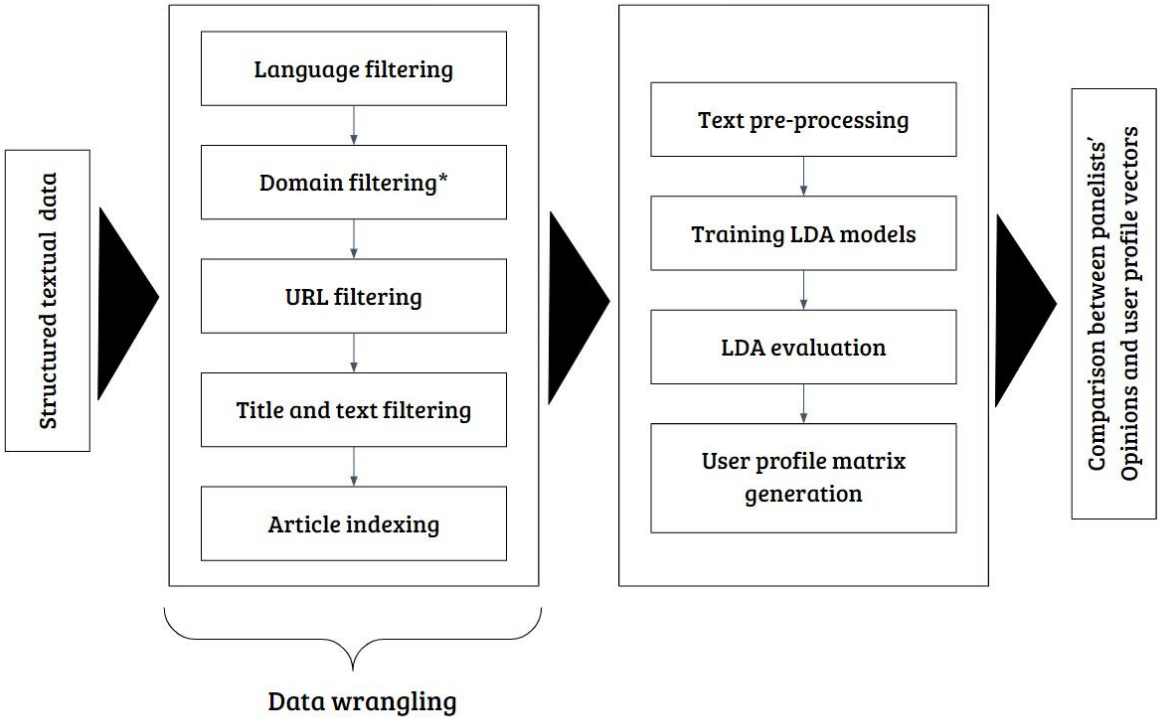


Figure 2 Graphical representation of thesis approach

2.1 Data Collection

The whole analysis relies mainly on self-selected samples of users who chose to engage with news items publicly. The data is based on their online article reading activity and their answers to a survey on their political opinions and their individual characteristics.

Web-tracking data

In order to construct an accurate picture of the news the panelists consume, we were provided with two web-tracking datasets that contain all of the links that each panelist visited within a manually curated list of UK and US news domains (table 1). The data also contains the parsed text from the scraped HTMLs in the form of a title and a text. We are interested in the following initial columns : title, text, meta_lang, meta_keywords, url, and panelist_id.

UK data = 'merged-parsed-tracking_uk.csv' | initial number of records = 676495
US data = 'merged-parsed-tracking_us.csv' | initial number of records = 248349

Survey data

Each participant is asked a set of questions about political issues, their social position and their salience perception.
Specifically, we are interested in the part where the panelists were asked separately about the issue they consider to be most important to them facing the United States or the United Kingdom at the moment. They were given specific options to choose from, as the table shows.

Issue
Crime
Economic situation
Rising prices - inflation - cost of living
Taxation
Unemployment
Terrorism
Housing
Government debt
Immigration
Health and social security
The education system
Pensions
The environment, climate and energy issues
Brexit: The decision of the United Kingdom to leave the European Union (only for UK)

Table 1 list of most important political issues that panelists have to choose from

2.2 Data wrangling

The purpose of our data wrangling pipeline is to create tidy data which can be used for analysis. This part of the thesis was the most challenging as the provided data included many texts that are not news articles, and there is no automated way of detecting them. We had to go through several steps and build multiple models to reach the desirable clean data.

Language filtering

The first thing to do was to get rid of articles that are not written in English.
After filtering on *meta_lang* column that supposedly keeps English articles only, each dataset still had a number of articles written in other languages, as seen in Table 2, which pushed us to apply the language recognition technique.

UK dataset	
Language	Count
English	590872
Italian	2329
French	144
Catalan	36
Scottish Gaelic	21
Spanish	16
Polish	15
German	8
Burmese	8

US dataset	
Language	Count
English	187730
Portuguese	220
Spanish	98
Dutch	84
German	60
French	11
Catalan	9
Basque	1
Japanese	1

Table 2 List of languages detected for both datasets

After a series of trials with different libraries, we opted for *fasttext* python library that is built on a pre-trained model "lid.176.bin" since it was the fastest and the most accurate. The function takes a sentence and predicts the languages with a certain confidence level using a neural network to incorporate words. We considered the language with the highest probability since most of the predictions were above 80% for the first language.

Domain filtering

In this section, we wanted to remove any URLs that are not included in the manually curated set of the admissible newspapers (domains).

Using that list alone was not effective enough in filtering the articles since we still had a lot of irrelevant pages. We have captured the presence of records that contain errors, logins, pop-ups, and mailboxes. An example of those records is shown in Table 3.

Subdomain	Text	Count
Polling	Not FoundThe requested URL / was not found on this server.	904
open.live	Not FoundThe requested URL / was not found on this server.Apache/2.2.25 (Unix) Server at open.live.bbc.co.uk Port 80	207
Careershub	Options for new users to register withYour social media account Linkedin Twitter FacebookOr with your email address	125

Table 3 Sample of irrelevant subdomains from bbc.co.uk and their respective texts

This part was **exclusively** applied to the UK dataset and not to the US because we have seen that the model performs very well on all news sites for the latter. This might be because the US newspapers structure and HTMLs are well organized and more focused on delivering the news than adding other irrelevant features.

The idea was to split each URL into a *domain*, *subdomain* and *tag* and manually check for the subdomains that do not include articles to get rid of them.

domain	subdomain
bbc.co.uk	[www, , careershub, news, bbcsignups.external,...
bloomberg.com	[www, , nav, login, personalization, careers, ...
cbsi.com	[popculture, chowhound, cbsnews, techrepublic,...
cbsnews.com	[www, , tealium]
dailymail.co.uk	[www, , discountcode, i, c-7npsfqifvt34x24ux2e...
businessinsider.com	[www, amp, markets, , static2, coupons, it, am...
cnn.com	[edition, www, us, amp, money, edition-m, edit...

Table 4 Subset of domains and their respective subdomains

This solution is time-consuming to do for all the domains. So, we figured to apply it only to the 20 most occurring newspaper domains in the UK dataset since they cover around 93% of records out of 176 present domains, as shown in Table 5.

domain	**occurrence**	**fraction**
bbc.co.uk	330264	0.556488
dailymail.co.uk	39864	0.067170
itv.com	29231	0.049254
sky.com	28566	0.048133
theguardian.com	23902	0.040274
thesun.co.uk	15383	0.025920
msn.com	15118	0.025474
bbc.com	10756	0.018124
telegraph.co.uk	10215	0.017212
express.co.uk	6513	0.010974
mirror.co.uk	6508	0.010966
newsnow.co.uk	5925	0.009984
inews.co.uk	4836	0.008149
independent.co.uk	4769	0.008036
metro.co.uk	4751	0.008005
thetimes.co.uk	4735	0.007978
channel4.com	3385	0.005704
indiatimes.com	3257	0.005488
livenewsnow.com	3214	0.005416
buzzfeed.com	2480	0.004179
	Sum Fraction:	**0.9287472**

Table 5 List of the top 20 most occurring newspapers and their dataset fraction

A result subset of our domain filtering approach applied on the top 4 UK domains can be found in Appendix A in the form of a table separating relevant subdomains and tags from irrelevant ones.

URL filtering

Another problem we encountered was that, when investigating the URLs in the dataset, we realized that many of them are main pages of different sections within newspapers. Table 6 and Table 7 show a subset of those pages and their respective counts.

For the UK data, we can see the main subsections of bbc.co.uk domain are dominating. It is because that particular domain makes more than 50% of all records, as seen in Table 5.

UK main section pages	Count
bbc.co.uk/news/uk	2891
www.bbc.co.uk/news/business	2553
www.bbc.co.uk/news/uk	2337
www.bbc.co.uk/news/england	2249
www.bbc.co.uk/news/world	1726
www.express.co.uk/	1689
www.bbc.co.uk/news/business/market-data	1611
www.bbc.co.uk/news/topics/c9qdqqkgz27t/ftse-100	1296
www.dailymail.co.uk/sport/index.html	1178

Table 6 An example of the main pages found in UK data

US main section pages	Count
www.foxnews.com/	6896
www.roughlyexplained.com/new-videos/	4313
www.nbcchicago.com/weather/maps/	2454
thehill.com/	1512
timesofindia.indiatimes.com/	1122
www.cbs.com/	1101
www.breitbart.com/	914
www.nbcnews.com/	857
www.fox9.com/	848

Table 7 An example of the main pages found in US data

We needed to get rid of those records as they contain pages that were recrawled many times, and the content shown has changed from the one the users saw the first time.

The first idea was to get the number of occurrences of each URL and get rid of any that are above a threshold, but we realized that, at any level, we cut off not just the main pages but also some highly viewed articles.

The technique that worked was to implement a function that parses the URL string and checks different conditions on the *suffix* for it to be a main webpage. It is based on our observation of the structure of URLs in our dataset. The function code is in Appendix B.

Text and title filtering

Before dropping the duplicates for our LDA model, we took our *text* and title columns separately. We computed the number of times they appear in our dataset using *exact string matching*. Then, we tried to filter out any records that are not articles by setting a cut-off threshold. The first one to check was the text column because it is the core of an article. So, if it is not an article, the title is unnecessary.
The determination of the ideal threshold for text filtering was made by computing different metrics gathered in Appendix C.

The procedure is as follows:

1. Create list of threshold $H \in \{500, 400, 300, 200, 100, 90, 80, 70, 60, 50, 40\}$ of maximum number of unique article's occurrences.

   We stopped at 40 because since we saw a huge information loss after that value which is something we are not interested in.

2. For **threshold** in H:
   - Get the list of unique texts that occur more than the threshold.
   - Count the number of relevant and irrelevant articles in that list.

- Compute **Data loss** as the number of unique, relevant articles lost at a certain threshold given the occurrences. For that, we created the function:

$$y_{data}(x) = \frac{t(x)}{T(x)}.u(x) \quad (5)$$

Where:

✓ $y(x) = information\ loss\ for\ threshold\ x.$

✓ $u(x) = number\ unique\ articles\ lost\ with\ the\ threshold\ x.$

✓ $t(x) = sum\ of\ lost\ relevant\ articles'\ occurrences\ with\ threshold\ x.$

✓ $T(x) = sum\ of\ lost\ articles'\ occurrences\ with\ threshold\ x.$

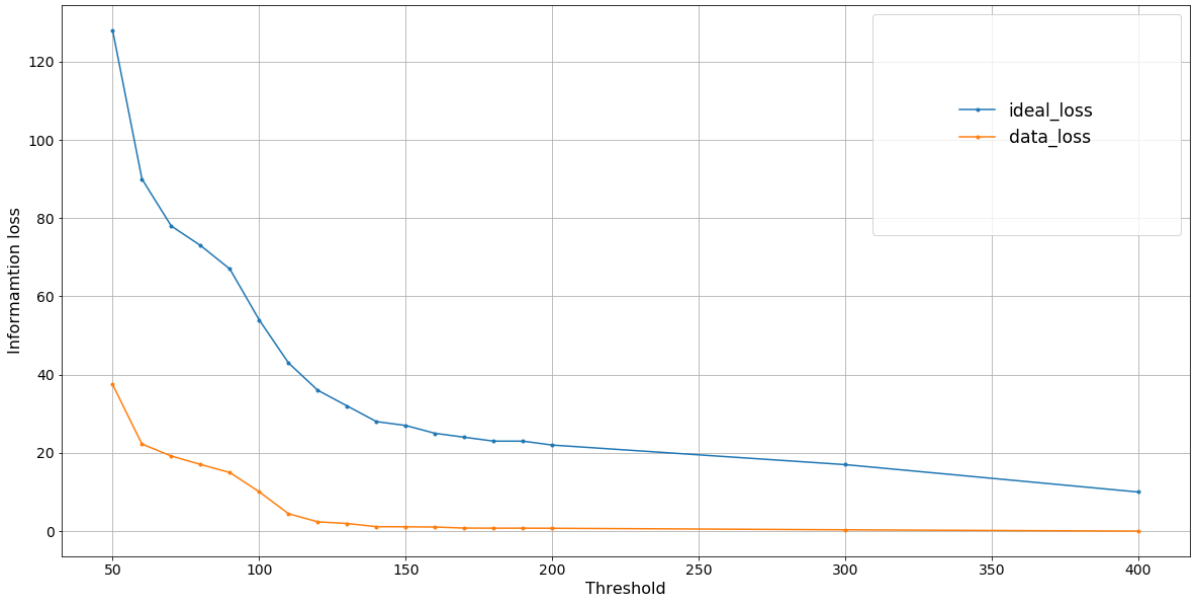
- Compute I**deal information loss** as $y_{ideal}(x)$ from the equation (5) by taking $\frac{t(x)}{T(x)} = 1$. The logic behind that value is that by taking a threshold x, all articles lost are relevant.
- Compute the **distance** using the following equation:

$$distance(x) = \left|\frac{y_{data}(x) - y_{ideal}(x)}{\max(H) - x + 1}\right| \quad (6)$$

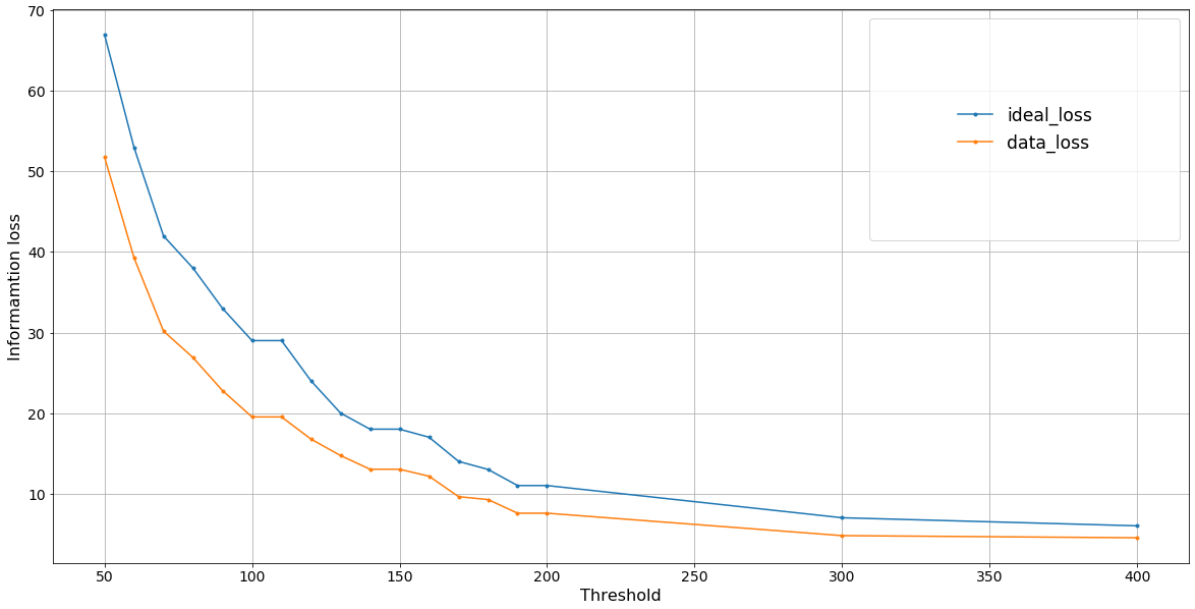
It is calculated as the absolute value of the difference between data loss and ideal information loss divided by the difference between the highest threshold and the current one. We added 1 in the denominator to avoid dividing by 0.

- Take the minimum distance as it is the minimum data loss across all thresholds.

Figure 3 and Figure 4 show the curves of both information loss and ideal information loss for different numbers of thresholds. As we can see, the gap expands when we take a lower threshold which makes sense knowing that texts with low occurrences have a high chance of being relevant. Hence, higher information loss when removing them.



(A) US dataset

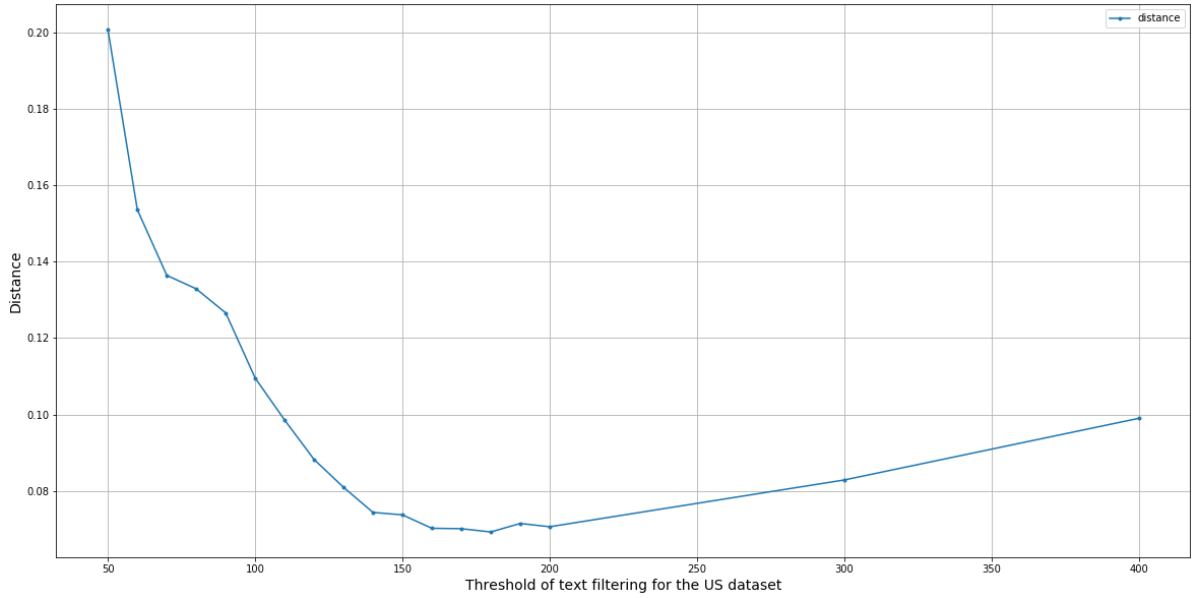


(B) UK dataset

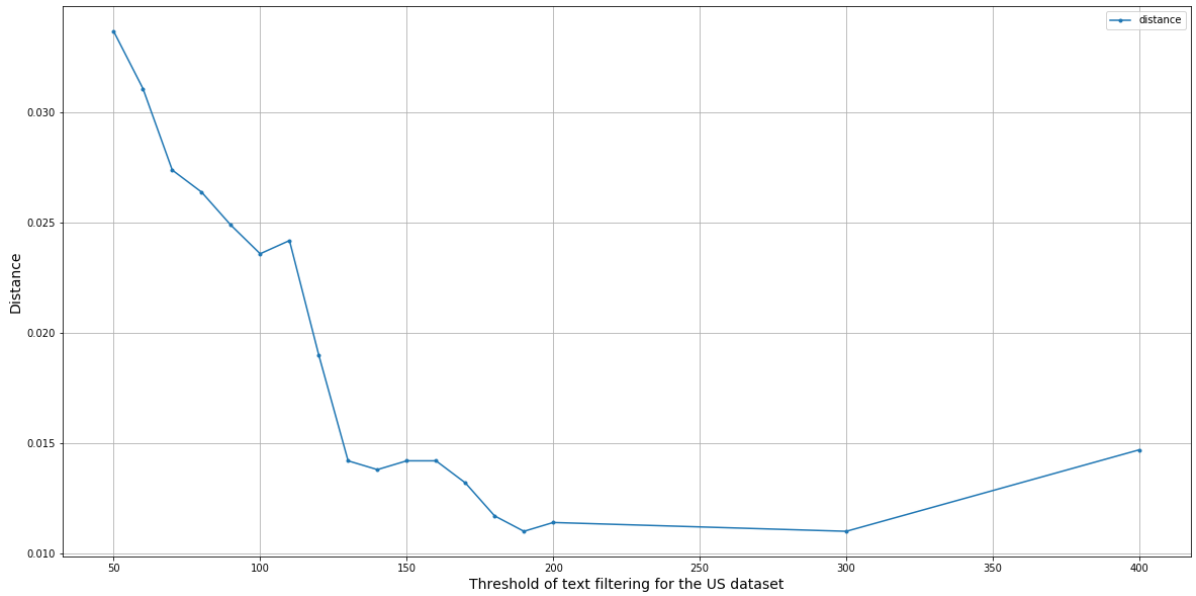
Figure 3 Number of unique, relevant articles lost for different thresholds in US and UK datasets

The difference in information loss between the US and UK data can be explained by the fact that the number of articles lost is much higher in the US dataset compared to its UK counterpart, even though the number of unique articles lost is low.

After computing our distance metric described in equation 6, we found the optimal threshold to be 190 and 180 for UK and US, respectively, as you can see in the following plots.



(A) US dataset



(B) UK dataset

Figure 4 Distance between data loss and ideal loss for different thresholds

After applying our thresholds to the data, we wanted to investigate the title column, but we realized that most of the titles we wanted to get rid of were gone. The ones that are left have a low number of occurrences. So, they will not affect the topic modeling.

This shows that combining both title and text from the beginning could have worked also. However, we were not sure about merging for the simple reason that we used exact string matching, so a simple difference in the combination can drastically change the result. This could be extended in the future to approximate matching using distance-based metric.

Article indexing

An article, in our case, is a string combination of *title*, *text*, and *meta_keywords*. We added formatted meta-keywords because they give an idea about the topics of the articles. In the list below that is a subset of the keywords we added and their number of occurrences in our dataset, we can clearly see that including these words will help to push the training to converge into giving human interpretable topics.

UK data	
Meta_keywords	count
Brexit	885
Politics	657
Theresa May	222
Donald Trump	217
US politics	127
Conservatives	102
Crime	83
Conservative leadership	81
Business	74
European elections	62

US data	
Meta_keywords	Count
Politics	977
US	823
Donald Trump	495
Entertainment	472
Health	323
Culture	318
Baseball	246
Business	177
Opinion	165
Abortion	104

Table 8 subset of the formatted keywords and their number of unique occurrences in our datasets

Later on, we are going to feed our model with the data after dropping the duplicates. Since one article could be read by different users, those deleted records need to be recovered to create the user topic profiles.

The solution was to add a new column to the datasets named *dup_index* in which each article and its duplicates will have a unique index. The algorithm is described below.

Pseudo-code for article indexing

Function indexing(x):
Input = x list of m strings
Output = list of unique indices of strings $list_{Index}$
Initialization: $list_{Index}, list_{Test}$ empty
For $i \in 1, \dots, m$:
    If i^{th} element of $x \notin list_{Test}$ **Then**
        If $i == 0$ **Then**
            Add 0 to $list_{Index}$
            Add i^{th} element of x to $list_{Test}$
        Else
            Add $[\max(list_{Index}) + 1]$ to $list_{Index}$
            Add i^{th} element of x to $list_{Test}$
        End
    Else add index of i^{th} element of x from $list_{Test}$ to $list_{Index}$
    End
End

Summary

After every stage of the data wrangling pipeline, we kept track of the records' numbers on each dataset. Table 9 summarizes it.

Steps	Remaining UK articles after step	Total UK data Lost	Remaining US articles after step	Total US data Lost
Initial data	676.495	-	248.349	
Keep English articles only	590.872	-85.623	187.730	-60.619
Domain filtering	141.676	-534.819	-	-60.619
URL filtering	88.955	-587.540	120.592	-127.757
Text/title filtering	85.081	-591.414	103.735	-144.614

Table 9 Data wrangling summary

The data wrangling got rid of almost 87% of the initial UK data and 60% of the initial US data. This approves our hypothesis in the beginning that UK data is noisier and needed an extra crucial part which is the domain filtering that took an initial 550.969 English articles and reduced it to 141.676 records.

2.3 Model training and evaluation

In this chapter, we are going to describe the topic model training step and the data manipulations needed for it. A general introduction to topic models and the LDA definition can be found in sections 1.1 and 1.2. Next, we will present the different results and evaluate them using the metrics introduced in section 1.3 to choose the best model for further analysis.

As mentioned in the previous chapter, this section starts by dropping all article duplicates. Hence, we are left with 27,830 Unique UK articles and 37,890 Unique US articles.

Data pre-processing

LDA is extremely dependent on the words used in a corpus and how often they occur. That is why we started working on normalizing our text using natural language processing techniques by creating a function that implements the following steps:

- Delete any handles, image captions, links, money amounts, phone numbers, and navigation menus that were still left from the HTML parsing.
- Change words to lowercase and delete characters other than letters.
- Tokenize the text by dividing it into words.
- Get rid of punctuation.
- Delete stopwords since these words are the most frequent, and they will have a negative impact on the model if included. We used a union of two English

stopwords dictionaries and a list of irrelevant words that show up in our topic keywords.

  - ✓ An example of the added words is in this sentence: "Media playback is unsupported on your device Media caption" that is present in 1819 of our UK articles. When crawling the URLs, some parts are not read in the HTML, mostly due to copyright reasons. Instead, a message is generated. The full list for both datasets can be found in Appendix D.

- Next, we use Spacy to extract the lemmatized forms of meaningful words. The first person replaces the words in the third person and the verbs of past and future tenses are changed to present.

Topic Modeling using LDA

To train our model, we must first form a dictionary from our corpus. We computed bigrams in the articles. They are sets of two adjacent words that occur many times together in the corpus (Ex. Email_address, affiliate_commission ...). Then, we mapped the corpus to word identifiers and converted the words using *a bag of words approach*. This process is called **textual representation**.

With everything ready for both UK and US datasets, we created 80 different LDA models by varying the number of topics (K) from 1 to 40 for each dataset. The choice of that range is based on the fact that news articles are mostly within 12 classes of topics (war, government, politics, education, health, the environment, economy, business, fashion, entertainment, sports, and unusual events). So, extending to a higher number of topics means going into sub-classes of those topics that could be merged.

We use the LDA model provided by Gensim. We needed to set values for the hyperparameters α and β. The choice was made based on the suggestion of (Yue Lu,

Qiaozhu Mei, & ChengXiang Zhai, Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA, 2010) that analyzed the choice of α with β and concluded that for topic classification, "the optimal performance is achieved when α is set to be small, e.g. between 0.1 and 0.5".

We set a value of $\alpha = 0.1$, meaning that each document is represented by fewer topics since we are interested in assigning each document to one topic and a low value of $\beta = 0.01$ in the interest of assigning fewer keywords to topics which will result in less "overlapping" between the latter. Then, we fixed random seed, which guarantees reproducibility.

LDA Model with K=20	
Topic #0: '0.104*"trump" + 0.041 *"president" + 0.020*"donald" + 0.014*"campaign" + 0.013*"white_ house"+ 0.009*"russian" + 0.008* "russia"+ 0.008*"tweet" + 0.008* "fbi" + 0.007*"obama"')	Topic #1: '0.038*"police" + 0.01 9*"man" + 0.013*"officer" + 0.01 3*"kill" + 0.012*"arrest" + 0.01 1*"shoot" + 0.010*"victim" + 0.0 09*"year old" + 0.009*"woman" + 0.009*"gun"'

Table 10 output sample of topics modeled in US data under Gensim implementation

However, at this point, the next questions we needed to ask ourselves were:

- **What number of topics (k) should we use to train our model?**
- **Can the generated topics easily be interpreted?**
- **Are the topics coherent?**
- **Is the purpose of the topic model fulfilled?**

To answer these questions in the next section, we used the set of evaluation metrics described in section 1.3.

Lda evaluation

In this part, we will be looking at the evaluation of the trained topic models and how it was done. The assessment was needed to quantify the quality of the generated topics.

Eyeballing

We started by inspecting the top $w = 10$ most probable words for each topic. The choice of w was based on the sum of probabilities corresponding to top $w \in [0,100]$ words in a topic by weight for different models. Figure 5 gives an example of this measure where it is clear that the top 5 to 20 ranked words in each topic are enough to capture its content due to the sharp drop in probability value from 0 to 20.

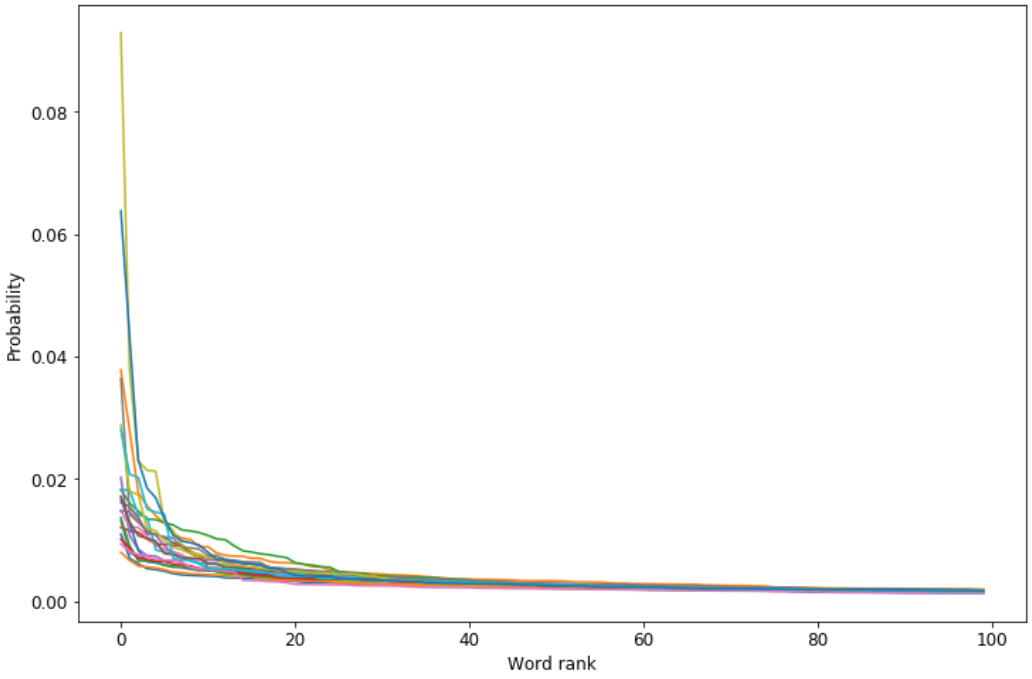


Figure 5 Probabilities of Top 100 Words in each Topic for K = 20 in the US data

This technique immensely helped us extend and improve the pre-processing part by adding meaningless words to the manually modified stopwords list.

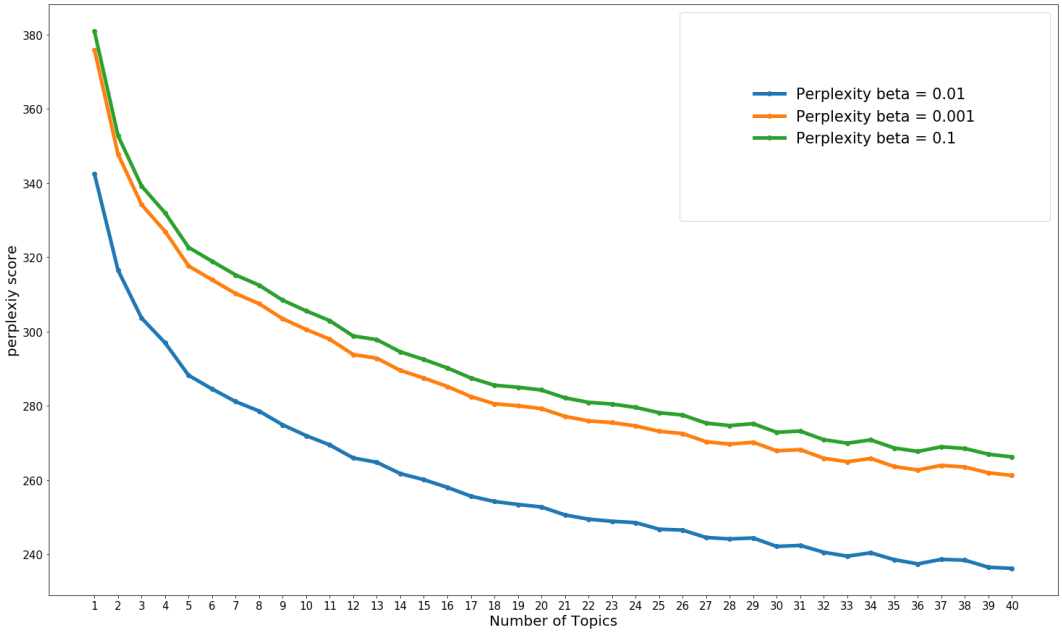
We also noticed that models with topics ranging between 15 and 25 for the US data and 12 to 20 for the UK data are the ones that cover most article topics. This observation could be biased since we worked with the data for a long time, and we know already the content of the articles. That is why we used the intrinsic evaluation metrics.

Perplexity score

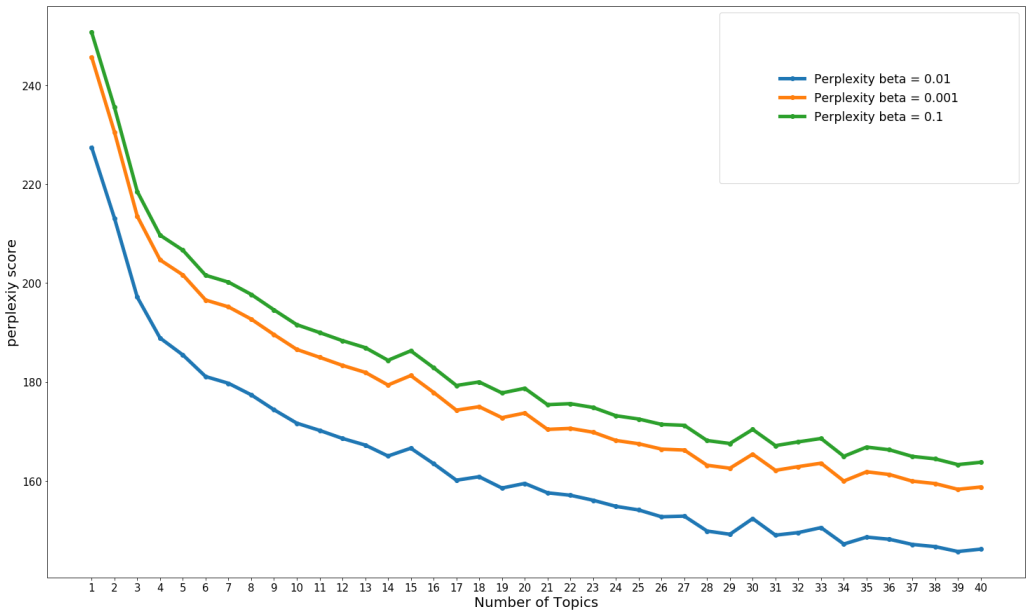
We mainly used this score to validate the choice of the LDA hyper-parameters that were suggested by Lu and al.

We trained multiple LDA models with fixed $\alpha = 0.1$ and β taking values from 0.001, 0.01 and 0.1. We then calculated the perplexity score of our models over the test set that is 1/300 data. The perplexity value p means that the model's confusion is as trying to guess the next word from p words. So, a lower value is better.

The plots below confirm the choice of $\beta = 0.01$ over the other values since it has the lowest perplexity scores over models with different numbers of topics.



(A) US data



(B) UK data

Figure 6 Perplexity scores as a function of the parameter beta and number of topics

Coherence score

To investigate the optimal number of topics discovered by our models, we computed U_{Mass}, UCI, and C_V scores introduced in section 1.3.2 for different numbers of topics (Appendix E). Then, we select a list of trained models with the admissible number of topics that maximize each score separately.

US DATA :

We started by analyzing UCI scores shown in Figure 7. The severe fluctuations are mainly due to a zoomed-in y-axis that is generally in $[-1,1]$ for this particular metric. The value kept improving with the number of topics till $K = 23$, where it reached its highest peak and then dropped drastically. This is a significant indication to choose the model with K = 23. The model with K= 22 was also promising since there is not much difference in UCI score between the two.

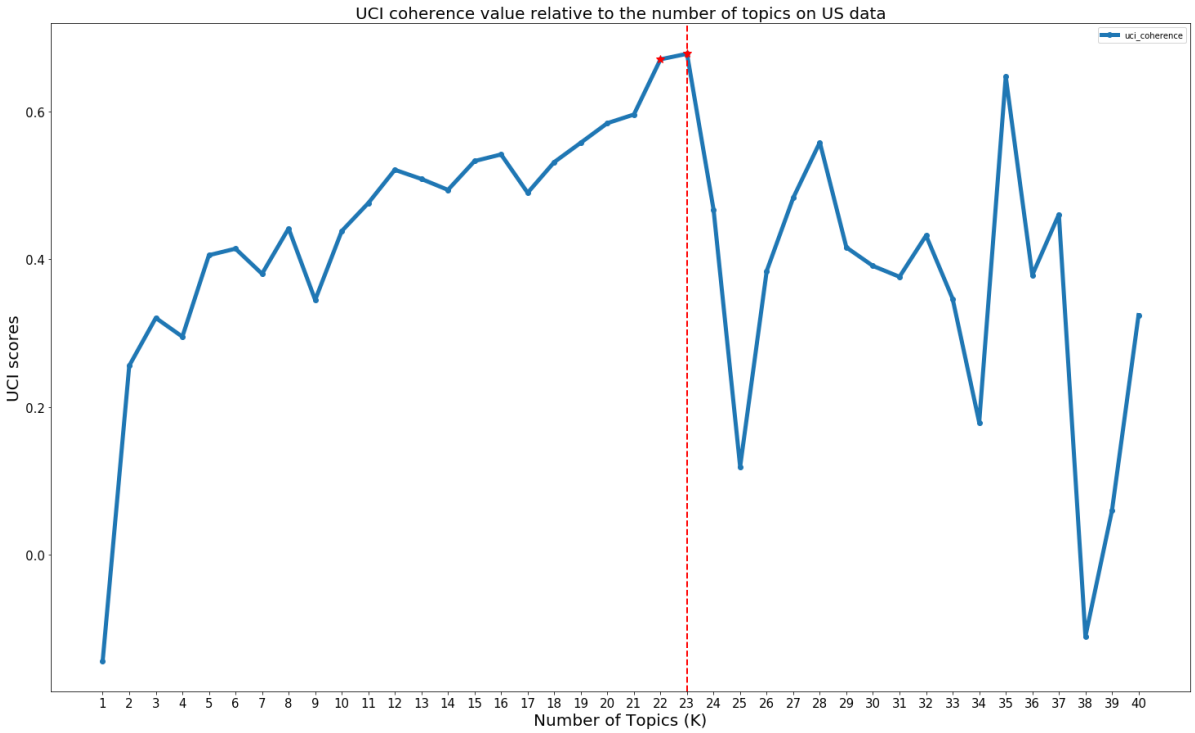


Figure 7 UCI coherence values relative to the number of topics on US data

Other peaks can be found in K= 28 and K= 35. The reason for not considering them is that when looking at the top 10 words of topics we found a lot of duplicated words across different topics. For example, for K=35, the word "trump" is found with high probability in 5 topics. We can understand that these models provide granular sub-topics due to the high number of topics.

Similarly, the highest value recorded for C_V score is when K=23. Figure 8 demonstrates that the rapid increase in value stopped around K=19. That is where it plateaued and kept fluctuating around $C_V = 0.58$.

The drop in value at K=9 made the investigation around that value interesting. However, not taking K that small is because the model does not capture the wide range of newspaper topics. As an example, the top keywords from one of the generated topics in the model with K= 8 (Figure 9) cover Immigration with word "border" and "mexico" and crime with "police", "court" and "arrest" which is not in our best interest.

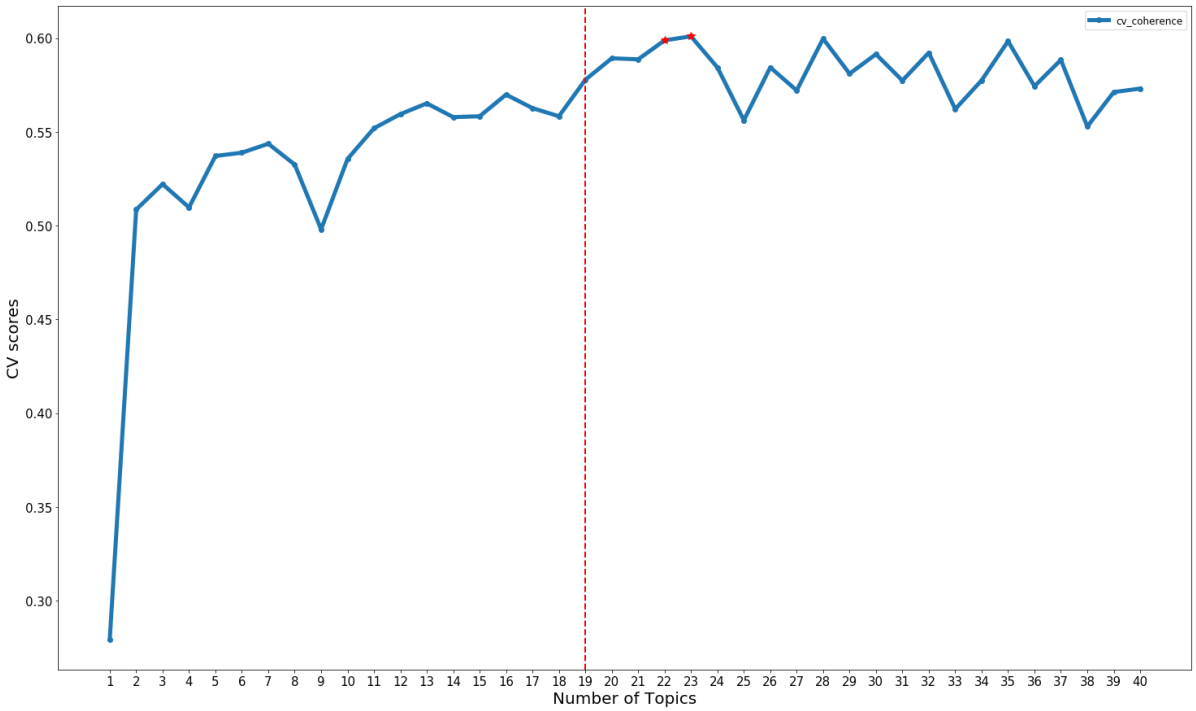


Figure 8 C_V *coherence values relative to the number of topics on US data*

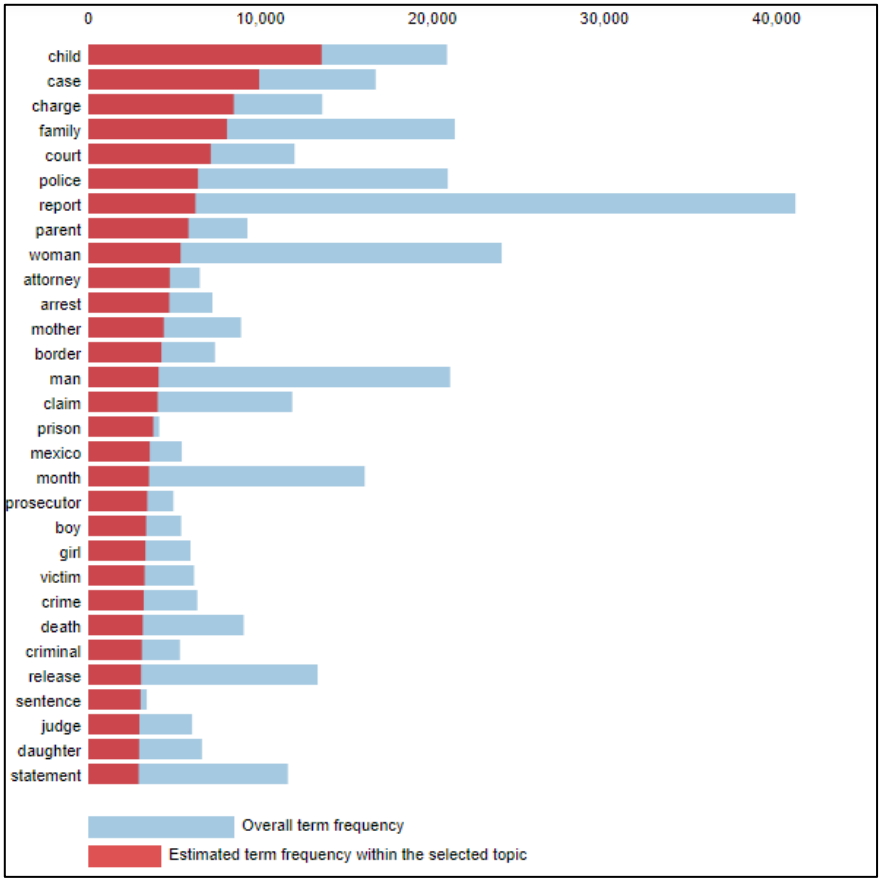


Figure 9 Top keywords of a random topic from model with K=8

The U_{Mass} plot in Figure 10 is a little bit tricky but still goes in the same logic. Usually, two criteria should be checked with this metric. The best model would be when the U_{Mass} value is in the interval where the values are relatively level, and the value should be the closest to 0. In our case, there are three stages where the value stabilizes across different numbers of topics. They are divided by the vertical red lines.

We are not interested in the first stage, where K between 6 and 13, as the number of topics is too small, as explained earlier. Also, stage 3 is too volatile to pick a certain K from it. The part where the value plateaus the best is for K between 19 and 22 before trending.

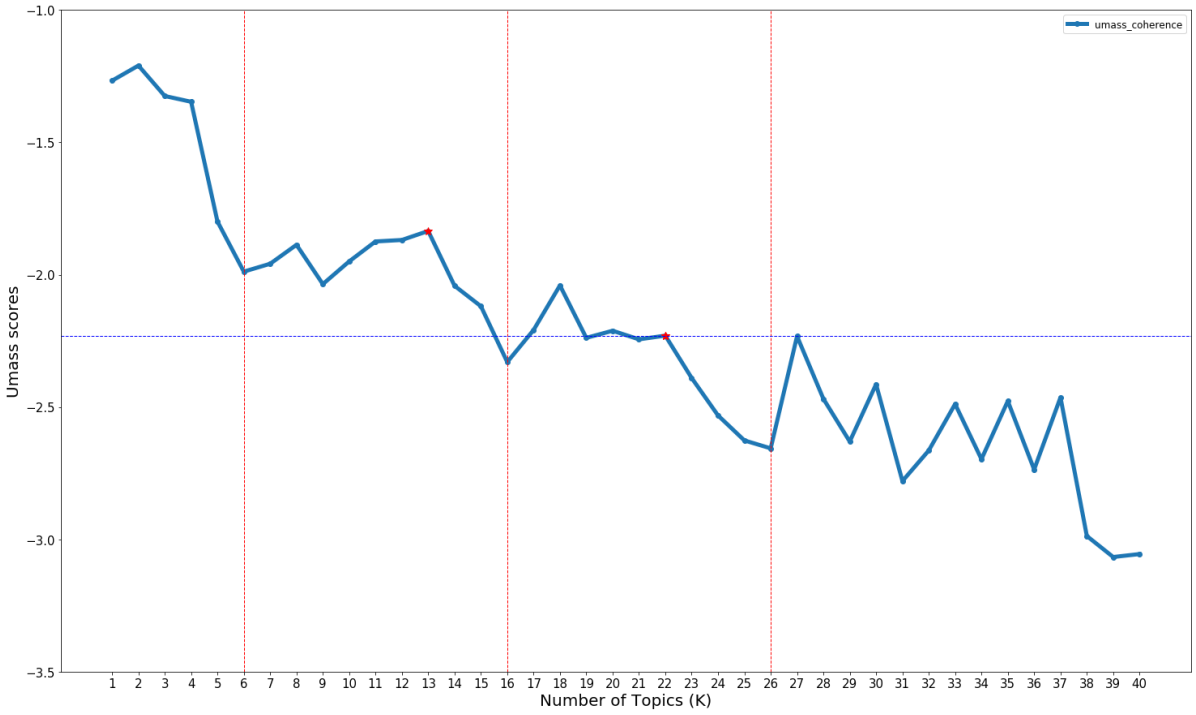


Figure 10 U_{Mass} coherence values relative to the number of topics on US data

Models with K= 22 and K= 23 are selected for further analysis.

UK data

In the case of UK data, the magnitude of fluctuations was much higher than the US one as shown in the plots of the coherence scores in Figure 11, Figure 12, Figure 13. So, the way to interpret those plots was by taking a set of best values from each score and find the intersection between them.

For UCI in figure 11, The score leveled up at K=8 and started fluctuating before trending down at K=26.

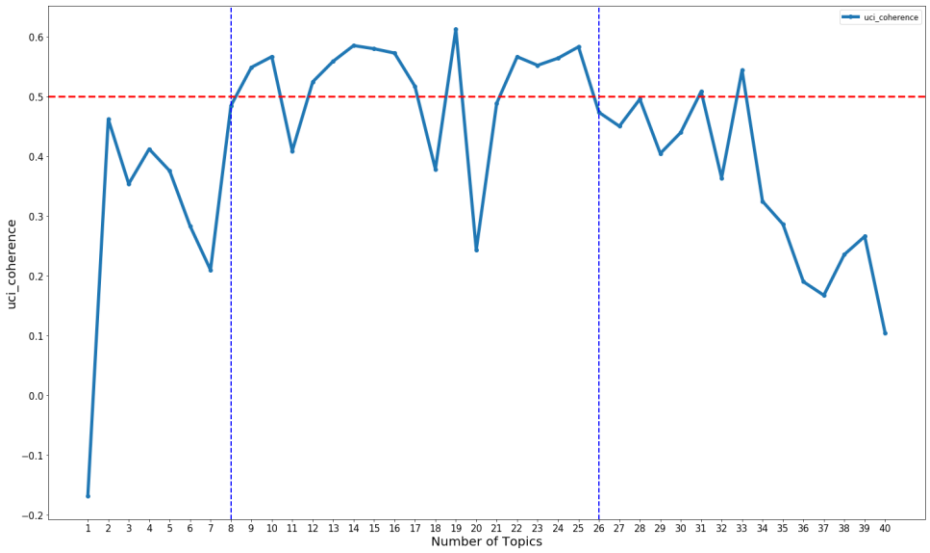


Figure 11 UCI coherence values for different number of topics on UK data

The best values recorded were for K= {10,14,15,16,19,22,25}

For C_V score in figure 12, the improvement stops around K=12, and, at K= 27, the scores started going downward.

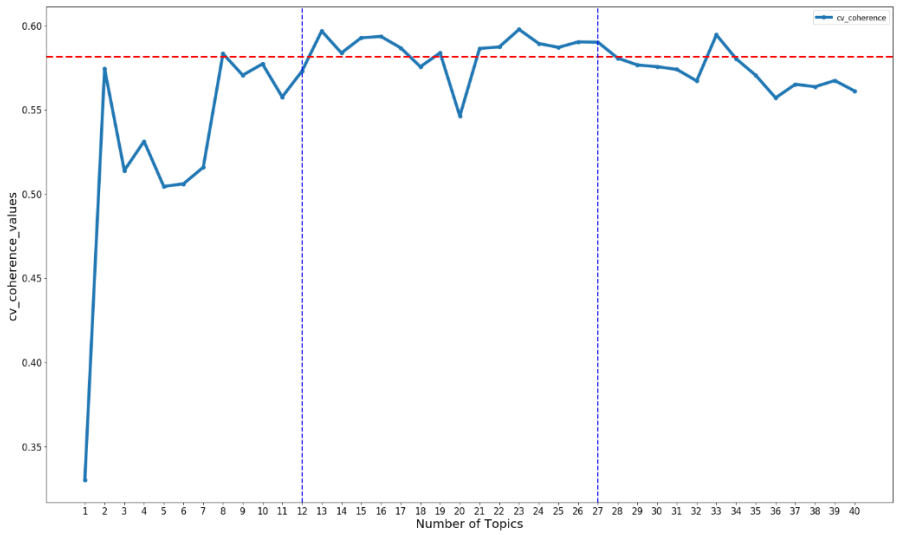


Figure 12 C_V coherence values for different number of topics on UK data

The best values recorded were for K= {13,15,19,23,27}.

As for the U_{mass} score shown in figure 13, The best values are in the range of K between 14 and 25 since this the area where the scores stabilized and varied around a constant U_{mass} =-2.5.

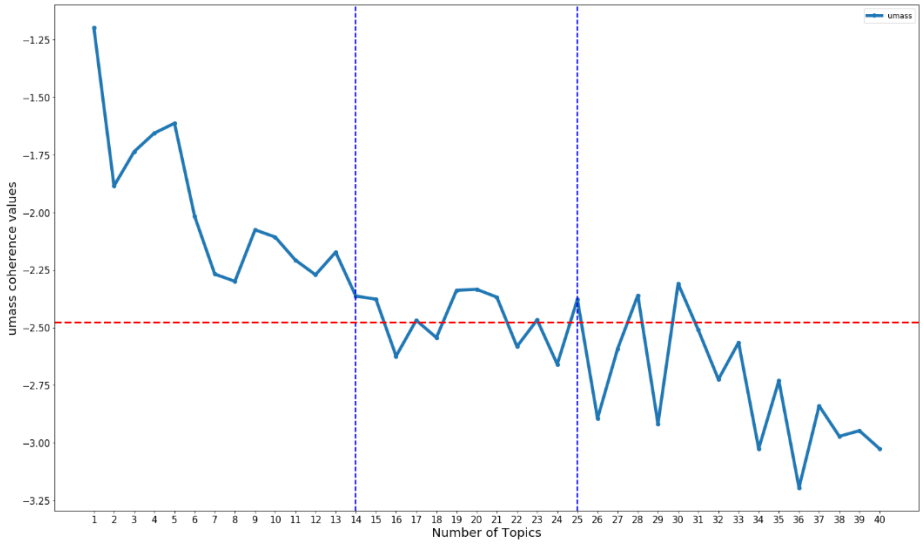


Figure 13 U_{Mass} coherence values for different number of topics on UK data

The best values recorded were for K= {14,15,17,19,20,21,23,25}.

The intersection between the best K values that optimize all coherence scores gives a selection of models that will be evaluated by the word intrusion technique.

$$\{10,14,15,16,19,22,25\}_{uci} \cap \{13,15,19,23,27\}_{C_V} \cap \{14,15,19,20,21,23,25\}_{U_{Mass}} = \{15,19\} \quad (7)$$

Models with K= 15 and K= 19 are selected for further analysis.

Word intrusion

In this part, we investigated the topics' interpretability by applying the word intrusion technique described in section 1.3.3.

We asked 10 participants that were not related to the analysis to find intruders in a list of 10 words per topic based on their interpretation. The input was the generated topics from the models we selected in the previous task. The models with $K \in \{22,23\}$ and $K \in \{15,19\}$ were to evaluate for the US and UK data, respectively.

The survey found in Appendix F contains four tables of generated topics from those models. Each row in those tables contains an intruder. The intruder was randomly selected from a topic where it has high probability and put in a topic where it has low

probability. The rows were randomly shuffled to prevent the participants from seeing a specific position pattern.

A score of 1 was assigned when an intruder was identified and 0 if not.

The answers are gathered in the following form:

Topic	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1	1	1	1	1	1	1	1	1	1	1
2	1	0	0	1	0	0	1	1	0	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	1	1	1	1
7	1	0	1	0	1	0	1	0	0	0
8	1	1	1	1	1	1	1	1	1	1
9	1	1	1	1	1	1	1	1	1	1
10	1	1	0	1	1	1	0	1	0	1
11	1	1	1	1	1	1	1	1	1	1
12	1	0	1	1	1	0	1	1	0	1
13	1	1	1	1	1	1	1	1	1	1
14	1	1	0	0	1	1	1	1	1	1
15	1	1	1	1	1	0	1	1	0	1
16	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	0	1
20	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1
22	1	1	1	1	1	1	1	1	1	1

Table 11 Answers of participants for LDA model with K=22 in the US data

We then averaged all scores on the number of topics which gave the model accuracy for a certain participant. After that, we averaged this measure again on the number of participants to get the global model accuracy.

Based on the results of that experiment displayed in Table 12, we can see that our participants' overall performance is outstanding in identifying the intruders within the provided sets of keywords even though they struggled with some specific topics where the answers were mainly wrong. This problem will be discussed later in the topic labeling part.

Number of topics / Participants	US Model K = 22	US Model K = 23	UK model K = 15	UK model K = 19
P1	1	0,913	0,866	0,947
P2	0,863	0,956	0,933	0,947
P3	0,863	0,956	0,866	0,789
P4	0,909	0,956	0,8	0,947
P5	0,954	0,956	0,933	0,894
P6	0,818	0,913	0,866	0,789
P7	0,954	0,956	0,8	0,894
P8	0,954	0,956	0,933	0,894
P9	0,727	0,956	0,866	0,842
P10	0,954	1	0,866	0,894
AVERAGE Model accuracy	0,9	0,952	0.87	0,884

Table 12 Results of word intrusion approach for model selection

This technique helped us incorporate human judgment in the evaluation of the LDA models' quality.

For what follows, we consider the model with K=23 for the US data and K=19 for UK data.

3 Result comparison

We came to the point where we need to answer our research question ***'How is the issue that a person considers most important related to what they browse?'***

We needed to label the topics we generated and create user profile vectors that will help us compare the topics browsed by the panelists with their preferred issue.

3.1 Topic labeling

In our attempt to make sense of the topics generated by the LDA models, we took samples of articles within each topic and tried to figure out the shared content between them.

- Result description for US data:

The list below shows each topic's top 10 keywords. We can see that most of these topics are coherent and easily interpretable.

Topic 0: ['star', 'love', 'film', 'work', 'play', 'hollywood', 'music', 'actor', 'fan', 'song']
Topic 1: ['border', 'mexico', 'immigration', 'migrant', 'wall', 'country', 'official', 'trump', 'immigrant', 'united']
Topic 2: ['house', 'senate', 'republican', 'vote', 'congress', 'gop', 'lawmaker', 'republicans', 'democrats', 'senator']
Topic 3: ['trump', 'president', 'military', 'country', 'official', 'israel', 'attack', 'war', 'government', 'leader']
Topic 4: ['company', 'work', 'service', 'online', 'amazon', 'offer', 'store', 'site', 'employee', 'business']
Topic 5: ['cathedral', 'notre_dame', 'church', 'paris', 'french', 'france', 'reuters', 'cat', 'animal', 'museum']
Topic 6: ['charge', 'case', 'court', 'attorney', 'prison', 'york', 'county', 'sentence', 'prosecutor', 'crime']
Topic 7: ['wear', 'royal', 'dress', 'queen', 'good', 'shoe', 'color', 'style', 'great', 'white']
Topic 8: ['trump', 'president', 'report', 'mueller', 'house', 'white', 'donald', 'special_counsel', 'investigation', 'congress']
Topic 9: ['child', 'family', 'life', 'mother', 'parent', 'woman', 'love', 'son', 'father', 'live']
Topic 10: ['food', 'restaurant', 'house', 'live', 'local', 'place', 'california', 'owner', 'resident', 'property']
Topic 11: ['study', 'human', 'plant', 'animal', 'climate_change', 'scientist', 'large', 'water', 'island', 'grow']
Topic 12: ['water', 'area', 'car', 'crash', 'park', 'air', 'flight', 'road', 'plane', 'hour']
Topic 13: ['woman', 'post', 'white', 'black', 'write', 'group', 'tweet', 'man', 'american', 'social']
Topic 14: ['fbi', 'investigation', 'campaign', 'report', 'russian', 'russia', 'official', 'clinton', 'email', 'department']
Topic 15: ['health', 'patient', 'medical', 'doctor', 'drug', 'study', 'eat', 'food', 'disease', 'treatment']
Topic 16: ['man', 'woman', 'story', 'video', 'night', 'room', 'hand', 'friend', 'watch', 'hear']
Topic 17: ['police', 'report', 'man', 'officer', 'kill', 'shoot', 'gun', 'death', 'arrest', 'victim']
Topic 18: ['law', 'abortion', 'court', 'rule', 'case', 'ban', 'supreme_court', 'woman', 'judge', 'legal']
Topic 19: ['game', 'season', 'win', 'team', 'play', 'player', 'final', 'character', 'episode', 'series']
Topic 20: ['school', 'student', 'high', 'university', 'college', 'teacher', 'parent', 'program', 'class', 'education']
Topic 21: ['pay', 'percent', 'tax', 'cost', 'billion', 'rate', 'high', 'trade', 'business', 'china']
Topic 22: ['trump', 'president', 'campaign', 'democratic', 'election', 'vote', 'biden', 'candidate', 'party', 'voter']

Table 13 List of topics with their top 10 keywords for US data with K=23

The interpretation and the labeling of topics is summarized below:

Topic 0: Celebrity: Famous people including musicians, actors and influencers.

Topic 1: Immigration: Sufferings of illegal immigrants and the actions made upon trump's executive decision to build a wall at borders with Mexico.

Topic 2: Political parties: Accusations between democrats and republicans to win votes for bills to pass.

Topic 3: International affairs: US foreign policy crises with Iran and North Korea.

Topic 4: Social media ads: Online business advertisements.

Topic 5: Disaster: Mostly about the major fire at Notre-Dame de Paris cathedral. It weirdly covers some rape cases including animal abuse. Hence, the presence of cat and animal in the keywords.

Topic 6: Crime: All kinds of crime that require sentencing from drug dealing, fraud, sexual assault to murder.

Topic 7: Lifestyle: News about royal family's lifestyle as well as style, grooming and fitness hacks.

Topic 8: Mueller investigation: Mueller's investigation into Russian interference in the 2016 US presidential election.

Topic 9: Family stories: happy and sad stories about people.

Topic 10: Restaurant reviews: overall reviews of restaurants including the food quality and the location.

Topic 11: Environmental issues: Research studies and warnings.

Topic 12: Weather Forecast: Warnings about thunderstorms, heavy rain, heat waves, and low visibility.

Topic 13: Racism: Racial discrimination, anti-Semitism, prejudice.

Topic 14: FBI and NSA cases: on different matters mainly on Hillary Clinton's email controversy.

Topic 15: Health: Information about Surgery Treatments, medical research, tips for better health.

Topic 16: Night stories: all kinds of mysterious stories about weird experiences.

Topic 17: Police: news about police getting injured, arresting people, and finding dead bodies.

Topic 18: Laws and bills: mainly about abortion bill across different states.

Topic 19: Sports: all kinds of sports.

Topic 20: The education system: anything related to school and university.

Topic 21: Finance: stock market, money management, housing and taxation.

Topic 22: Presidential campaign: the race between Trump and Biden to win the elections.

We can already see that some of the topics can be clustered together, making a higher class. A clear example would be **topic 14** with **topic 8,** which are both about investigations. Later on, we are going to discuss those clusters in depth.

We also noticed that some of the topics are specific to certain periods. Topic 18 and topic 8 are clear examples of that observation. This gives an insight to further analyze the topics with respect to time periods in future work.

Figure 14 shows the initial US document distribution over the generated topics. The top 3 most important articles for the US readers cover law enforcement, the Mueller investigation of Russians interfering with the US elections of 2016 and the presidential race. This makes sense since these were hot topics in the time frame of data collection.

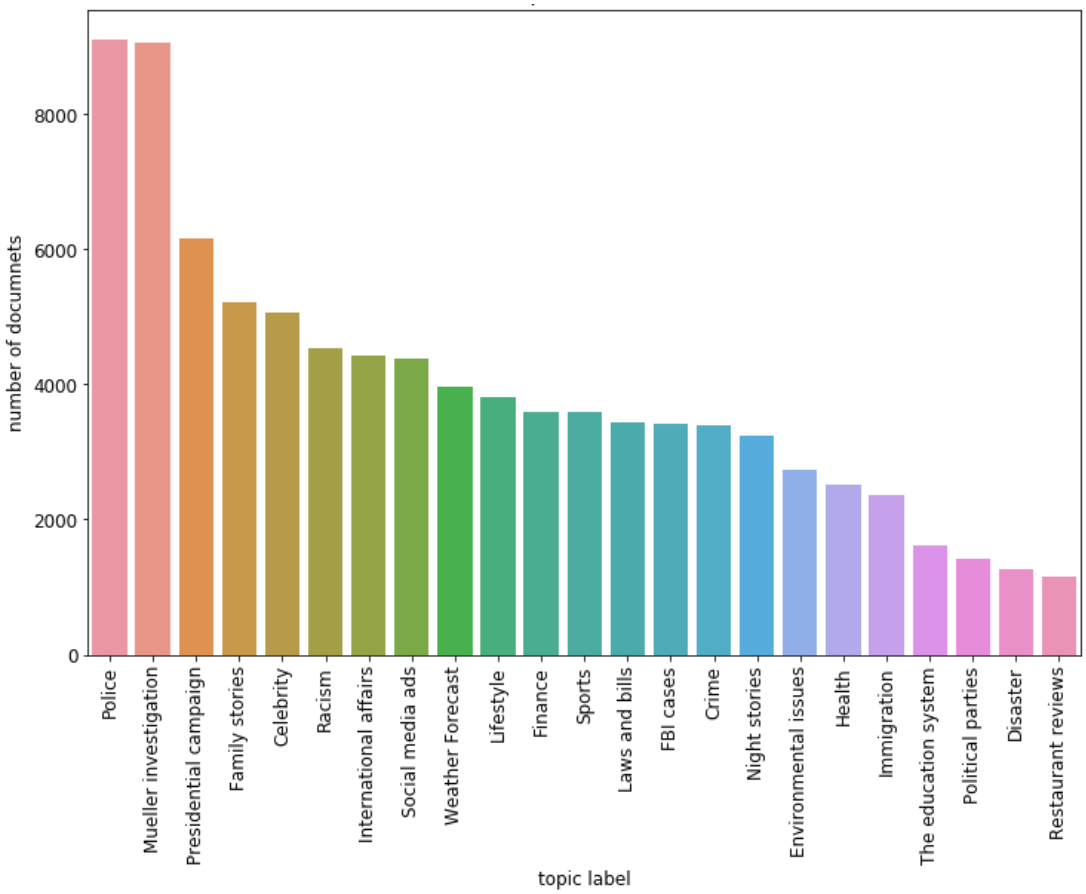


Figure 14 US topic distribution

- Result description for UK data:

In the same fashion, we are interested in labeling the UK topics. The list below reveals the top 10 keywords for each topic generated by our LDA model for K = 19

Topic 0: ['brexit', 'party', 'deal', 'vote', 'eu', 'labour', 'election', 'tory', 'mps', 'leave']
Topic 1: ['credit', 'flight', 'plane', 'fly', 'beach', 'air', 'passenger', 'crash', 'airport', 'animal']
Topic 2: ['trump', 'president', 'country', 'war', 'government', 'force', 'military', 'leader', 'china', 'donald_trump']
Topic 3: ['royal', 'queen', 'baby', 'meghan', 'harry', 'prince', 'wedding', 'birth', 'family', 'visit']
Topic 4: ['case', 'court', 'claim', 'report', 'law', 'charge', 'investigation', 'woman', 'public', 'legal']
Topic 5: ['wear', 'dress', 'black', 'star', 'pair', 'white', 'instagram', 'love', 'daughter', 'couple']
Topic 6: ['police', 'man', 'attack', 'officer', 'woman', 'arrest', 'car', 'road', 'incident', 'kill']
Topic 7: ['work', 'school', 'government', 'child', 'uk', 'university', 'student', 'support', 'high', 'report']
Topic 8: ['feel', 'woman', 'write', 'man', 'post', 'twitter', 'social', 'good', 'lot', 'life']
Topic 9: ['win', 'game', 'play', 'team', 'player', 'city', 'final', 'match', 'england', 'football']
Topic 10: ['cathedral', 'paris', 'notre_dame', 'church', 'french', 'france', 'hotel', 'family', 'century', 'king']
Topic 11: ['council', 'car', 'house', 'local', 'city', 'build', 'park', 'street', 'work', 'road']
Topic 12: ['company', 'business', 'customer', 'price', 'store', 'uk', 'firm', 'sell', 'buy', 'product']
Topic 13: ['pay', 'bank', 'club', 'chelsea', 'season', 'liverpool', 'money', 'united', 'tax', 'arsenal']
Topic 14: ['health', 'patient', 'hospital', 'doctor', 'body', 'eat', 'woman', 'food', 'drug', 'treatment']
Topic 15: ['family', 'child', 'mother', 'life', 'death', 'leave', 'court', 'daughter', 'son', 'month']
Topic 16: ['protest', 'reuters', 'protester', 'extinction_rebellion', 'activist', 'climate_change', 'group', 'demonstrator', 'police', 'parliament']
Topic 17: ['weather', 'south', 'service', 'east', 'water', 'west', 'train', 'uk', 'monday', 'blaze']
Topic 18: ['star', 'film', 'love', 'play', 'fan', 'series', 'tv', 'actor', 'leave', 'life']

Table 14 List of topics with their top 10 keywords for UK data with K=19

For each topic, we read a subset of articles and kept expanding it till we capture the class of articles. The interpretation is shown as a label and a small description of its content.

Topic 0: Brexit: anything related to UK's withdrawal from the European Union.

Topic 1: Air travel: anything related to flights, airplane crashes and weather.

Topic 2: International affairs: US concerns with Iran and North Korea actions.

Topic 3: Royal Family: News about the UK royal family (marriages, visits…).

Topic 4: Crime: court cases and sentencing scenarios.

Topic 5: Celebrity: discusses the lifestyle and fashion of celebrities.

Topic 6: Police: law enforcement interventions.

Topic 7: Education: discusses the education issues and UK government's efforts to b etter the situation.

Topic 8: Social stories: people telling stories about social situations like dating.

Topic 9: Sports: mainly Premier League (football).

Topic 10: Disaster: the major fire at Notre-Dame de Paris cathedral.

Topic 11: Construction: new buildings, renovation, car tickets for invading construc tion sites.

Topic 12: Business: company activities, introduction of new technologies, closing of stores.

Topic 13: Football transfers: contract signing, salary raise.

Topic 14: Health: information about Surgery Treatments, medical research, tips for b etter health.

Topic 15: Domestic violence: abuse, murder within families.

Topic 16: Environmental issues: News about climate change and protests made by Extinction Rebellion which is an environmental social movement.

Topic 17: Weather forecast: mostly articles from the Met Office that explains the dire ction of the storms and the delay in trains caused by disruptions

Topic 18: TV production: discusses the news of actors and actresses taking new role s, movie and TV show reviews

Similar to the US data, most of the extracted labels are closely related to the keywords shown. Figure 15 shows the document distribution over the labeled topics.

The UK article's readers were most interested in Brexit news. The latter makes twice as big as the closest topic, "Tv production" in volume. This gap can change when merging topics that are of the same class.

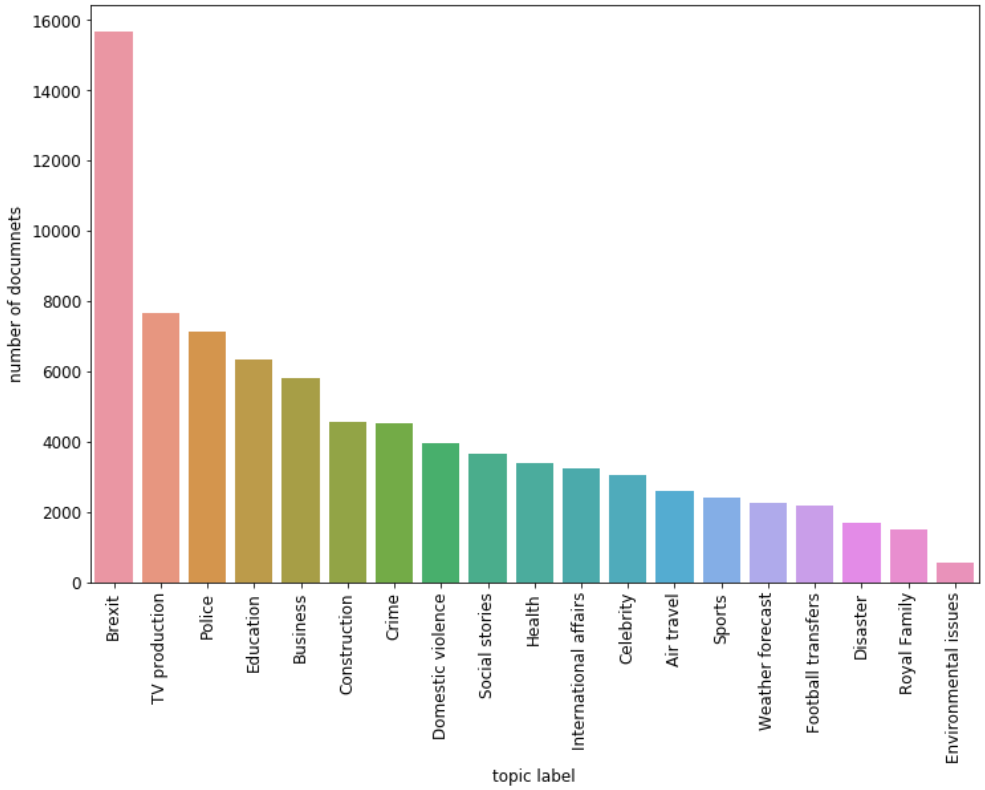


Figure 15 UK topic distribution

3.2 User profile vectors

In this part, we will be creating user profile vectors based on the datasets used in training.

By default, each article is a weighted combination of all the topics in the LDA implementation as shown in section 1.2. In our analysis, we assign an article to the topic with the highest probability.

We define $a_{i,j}$ as a fraction of articles read by $panelist_i$ under a certain $topic_j$:

$$a_{i,j} = \left(\frac{number\ of\ articles\ read\ by\ panelist_i\ under\ topic_j}{number\ of\ articles\ visited\ panelist_i} \right)$$

The profile vector $\overrightarrow{p_i}$ for $panelist_i$ is then:

$$\overrightarrow{p_i} = (a_{i,0}, a_{i,1}, \ldots . a_{i,(K-1)}) \text{ where } \sum_{j=0}^{K-1} a_{i,j} = 1$$

An example is shown in Figure 16, panelist with id= 00a3c224cc20373a did not read any articles from the topics 0, 2, 4, 5, 6, 7, 8, 9…

panelist_id	0	1	2	3	4	5	6	7	8	9	...	22
006e503ead8eb405	0.007838	0.294462	0.031905	0.000000	0.000000	0.000000	0.042782	0.000000	0.000000	0.000000	...	0.043702
00a3c224cc20373a	0.000000	0.017932	0.000000	0.045129	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.074925
00c2347435c7d477	0.000000	0.000000	0.000000	0.141783	0.117639	0.000000	0.000000	0.000000	0.000000	0.000000	...	0.000000
00c515cab74b184a	0.000000	0.000000	0.014525	0.033819	0.217923	0.002725	0.016753	0.060728	0.071353	0.000000	...	0.030550
00d171d9c44c9268	0.001699	0.002113	0.017745	0.022825	0.096985	0.009095	0.024535	0.098544	0.008998	0.000000	...	0.100876

Figure 16 profile vectors in US dataset with K=23

Each panelist has a weight for every topic and the highest one is the one we compare With the preferred issues discussed in 2.1.

3.3 Issue and user profile comparison

Now that all elements are between our hands, we are going to start by adding all the variables together in a dataframe.

We retrieved panelist_id along with issuesfirstW1 and issueseufirstW1 from the survey for US and UK data, respectively. The number of panelists left in our UK data is 782. As for the US data, 1061 panelists were left for the analysis.

The final dataframe contains for each user:

- The profile vector across all the topics.
- The label of the top browsed topic by the user.
- The global issue that the user finds the most important.

We were interested in comparing:

- Survey opinions from the user profile vector when aggregating users based on Top topics. (variable 1)

- Top topics from the user profile vector when aggregating users based on survey opinion. (variable 2)

For that, we created a subset of labeled topics for both the UK and US data and a subset of issues that we figured could give a good base for discussion since some of the labels are not matching with the predefined issues.

The lists below show the chosen labels for each dataset to compare.

Predefined Issues
Crime
Immigration
Health and social security
The education system
The environment, climate and energy issues
Brexit

US Labels
Crime
Immigration
Health
The education system
Environmental issues

UK labels
Crime
Health
Education
Environmental issues
Brexit

We computed bar plots of variable_1 when aggregating users based on the labels and variable_2 when aggregating users based on the issues.

Those plots are found side by side in appendix G.

The blue plots correspond to the number of panelists for which an issue (x-axis) is the most important. In the meantime, the red plots correspond to the number of panelists for which a labeled topic was the most read.

3.4 Results

For US data:

The survey answers are mostly favoring Health and Immigration for the analysis since both have high occurrences as the most important issue in our dataset, as shown in the table below

issuesfirstW1	Count
Health and social security	204
Immigration	201
Rising prices / inflation / cost of living	104
Economic situation	85
Terrorism	82
Other	78
The environment, climate and energy issues	70
Crime	62
Don't know	44
Government debt	37
The education system	29
Taxation	23
Unemployment	20
Housing	16
Pensions	6

The most important findings are the comparison between plots for:

- "**Health/issuesfirstW1**" and "**Health and social security/US label**" :

  The highest bar for the blue plot corresponded to the predefined issue Health and social security. It means that people reading "labeled Health" are mostly the ones that said Health is the most important issue to them. When comparing this value to the red plot of "Health and social security/US label", we found that labeled Health was still the highest amongst the different generated topics we selected. We neglect the labels that are not considered in the subsets since they are irrelevant to the comparison.

- "**Immigration/issuesfirstW1"** and "**Immigration/US label**" :

  The highest bar for the blue plot corresponded to the predefined issue Immigration. So, people who read articles labeled Immigration are mostly the ones that consider that topic to be the important issue. When comparing this value to the red plot of "Immigration/US label", we saw that our labeled Immigration comes at the 3rd position in our subset and that the people who considered that issue as the most important read more about Health and Crime.

- "**Environmental issues/issuesfirstW1**" and "**The environment, climate and energy issues/US label**". Another interesting plot where the environment, climate and energy issues jumped to the 2nd position after Health. The value stayed high also for the labeled topics in second position after Health which has a high occurrence in our dataset as shown before and also a related topic to the environment. This shows, even though the number of participants considering it as the most important is low, they stuck with their choice.

The findings that raise questions are the ones for "**Crime**" and "**The education system" issues**. The plots show people who said those topics are the most important issues are not interested about them in their reading even though they are part of our model's best generated topics.

For UK data:

Looking at the blue graphs in Appendix G (continued), we see a trending high peak for issue Brexit which is because most participants voted Brexit as the most important.

issueseufirstW1	Count
The decision of the United Kingdom to leave the European Union (Brexit)	411
Health and social security	56
The environment, climate and energy issues	51
Immigration	47
Crime	38
Rising prices / inflation / cost of living	33
Housing	27
Economic situation	27
Don't know	27
Terrorism	20
Other	20
The education system	8
Government debt	6
Unemployment	5
Pensions	3
Taxation	2

The gap is even higher when the labeled topic is also Brexit. The positive correlation is validated by the red plot "The decision of the United Kingdom to leave the European Union (Brexit)/UK label".

For other issues, there isn't much to say when comparing these metrics for the UK dataset since the opinions are highly skewed towards one value "Brexit".

3.5 Discussion

From the results described in the previous section, we can only say that answering our research question using our analysis alone is not enough. Some of the labeled topics in our corpora gave promising results while others did not.

A more complex task is needed in the labeled topics / issue aggregation. For example, this can be achieved by creating a set of keywords that covers each topic's content from pre-labeled articles and then make clusters of topic classes based on those keywords.

This idea comes to mind when looking, for example, at the set of topics [Crime, Police, and Investigation] that can all go under one class "Crime".

Conclusion

In the present work, we provided different techniques to prepare the data for model training and optimize the LDA output. The approach consisted of first creating a new evaluation technique to detect meaningful and relevant textual articles in a corpus. Another technique introduced was in the model evaluation, where we combined four existing intrinsic metrics to come up with the best model for training.

When trying to see if a participant mostly reads articles that he thinks are strongly related to what he prefers, some of the findings validated that assumption while others did not. This is where the limitations of our approach showed up.

This work could be extended later into applying multi-labeling topic modeling where we assign different topics to a text document. Another idea could be to explore the topics for different time frames.

The source code used for the implementation of all the steps can be found in the following Github repository :
https://github.com/gesiscss/Information-News-exposure-issue-salience

Appendix A

Example of the domain filtering technique on the top 4 UK domains

domain	subdomain	Relevant/Irrelevant
bbc.co.uk	'' , 'www'	Relevant with the tag /news/
	'careershub', 'account', 'bbcsignups.external', 'm' 'session', 'ssl', 'careers', 'searchclick', 'swscdn', 'bbcgoodfood', 'shop', 'downloads-app.iplayer.api', 'search', , 'click.email', 'beta', 'iplayerhelp.external'	Irrelevant
theguardian.com	'ablink.editorial'	Relevant
	'', 'www'	Relevant with tags : /news/world /commentisfree/us-news/uk-news/ politics/education/society/science/ business/money/sport/ technology/ australia-news/travel/commentisfree/
dailymail.co.uk	'', 'www'	Relevant with tags : /news/debate/ columnists/ femail/health/money/ sport/travel/tvshowbiz/
	'c-5uwzmx78pmca09x24ntqx78jwizlx2ekwu.g00', 'c-7npsfqifvt34x24gbwfx2edp.g00', 'fff', 'c-7npsfqifvt34x24x78x78x78x2etjmlgsfex2edpn.g00', 'c-6rtwjumjzx7877x24bbbx2emtrjx78jwajx2ehtr.g00', 'c-6rtwjumjzx7877x24fihqnhpx2elx2eitzgqjhqnhpx2esjy.g00','click.email.discountcode', 'gotolink', 'creative', 't', 'secured', 'discountcode	Irrelevant
sky.com	'news'	Relevant with tag /story/
	'', 'ogwam', 'www', 'epgservices', 'helpforum', 'secure', 'buy', 'messages', 'mysky', 'my', 'trackmyorder', 'pubfinder', 'tv', 'myhelprequests', 'skyvision', 'broadbandshield', 'go', 'payments', 'election.news', 'partner.help','customermonth', 'careers', 'myaccount', 'contactus', 'broadbandreconnection', 'rewards', 'claim', 'www.fafe', 'community'	Irrelevant

Appendix B

Python Code for URL filtering

```
1.  def check_url(url):
2.      check = url.split("/")[1:]
3.
4.      if "index.html" in url :
5.          return(True)
6.      elif (url[-1] == "/"):
7.          if len(check) <= 1 :
8.              return(True)
9.          elif (len(check[-2])< 15) and (check[-2].isnumeric() == False)  :
10.             return(True)
11.     elif len(check) == 0 :
12.         return(True)
13.     elif (len(check[-1])< 15) and (check[-
    1].isnumeric() == False) and ("?" not in url) and ("html" not in url):
14.         return(True)
15.     else :
16.         return(False)
```

Appendix C

Resulted table from Text filtering approach for US and UK datasets

US Dataset						
threshold	relevant articles lost	total articles lost	unique articles lost	data loss	ideal loss	distance
500	0	12783	9	0.0	9	-
400	0	13212	10	0.0	10	0.099
300	313	15633	17	0.34	17	0.0829
200	564	16857	22	0.736	22	0.0706
190	564	17056	23	0.761	23	0.0715
180	**564**	**17056**	**23**	**0.761**	**23**	**0.0693**
170	564	17226	24	0.786	24	0.0701
160	728	17390	25	1.047	25	0.0702
150	728	17699	27	1.111	27	0.0738
140	728	17844	28	1.142	28	0.0744
130	1125	18374	32	1.959	32	0.081
120	1245	18864	36	2.376	36	0.0883
110	2037	19656	43	4.456	43	0.0986
100	3878	20802	54	10.067	54	0.1096
90	4923	22034	67	14.97	67	0.1266
80	5267	22548	73	17.052	73	0.1329
70	5644	22925	78	19.203	78	0.1364
60	5857	23684	90	22.257	90	0.1536
50	7524	25711	128	37.458	128	0.2008

Appendix C (continued)

Resulted table from Text filtering approach for US and UK datasets

UK dataset						
threshold	relevant articles lost	total articles lost	unique articles lost	data loss	ideal loss	distance
500	0	751	1	0.0	1	-
400	2281	3032	6	04.514	6	0.0147
300	2281	3340	7	04.781	7	0.0111
200	2978	4329	11	07.567	11	0.0112
190	**2978**	**4329**	**11**	**07.567**	**11**	**0.0110**
180	3342	4693	13	09.258	13	0.0117
170	3342	4864	14	09.619	14	0.0132
160	3826	5348	17	12.162	17	0.0142
150	3984	5506	18	13.024	18	0.0142
140	3984	5506	18	13.024	18	0.0138
130	4250	5772	20	14.726	20	0.0142
120	4376	6265	24	16.764	24	0.0190
110	4604	6839	29	19.523	29	0.0242
100	4604	6839	29	19.523	29	0.0236
90	4986	7221	33	22.786	33	0.0249
80	5409	7644	38	26.889	38	0.0264
70	5705	7940	42	30.178	42	0.0274
60	6408	8643	53	39.295	53	0.0311
50	7270	9405	67	51.791	67	0.0337

Appendix D

Irrelevant words in corpus

It was created for the purpose of deleting meaningless words that appear in different generated topics for both datasets.

reaction	mr	affiliate_commission	ve	lee
copyright	bbc	didn	nz	los_angeles
media	image	share	reaction	bu
playback	de	september	copyright	nz
unsupported	en	long	media	reaction
device	caption	reuters_slide	playback	copyright
afpgetty	also	jones	unsupported	media
slide	copyright	affiliate_commission	device	playback
people	something	twitter	purchase_recommend	unsupported
day	nh	facebook	ms	device
news	getty	august	dog	link_article
fl	pa	october	afpgetty	ms
pause_gif	don	jo	microsoft_earn	dog
pic_twitter	ap	thoma	credit	turn
include	afp	june	san_francisco	inch
cbs	reuter	july	city	bu
james	picture	fox	time	one
april	april	james	feel	reaction
include	may	photo	cbs	copyright
Dog	day	april	london	news

Appendix E

Coherence values for US and UK datasets over different numbers of topics

US data				*UK data*		
Number of topics	U_{Mass}	C_V	*UCI*	U_{Mass}	C_V	*UCI*
1	-1.267	0.279	-0.144	-1.199	0.33	-0.168
2	-1.21	0.509	0.256	-1.885	0.574	0.462
3	-1.325	0.522	0.321	-1.736	0.514	0.354
4	-1.347	0.51	0.295	-1.656	0.531	0.412
5	-1.799	0.537	0.406	-1.613	0.504	0.376
6	-1.988	0.539	0.414	-2.017	0.506	0.283
7	-1.959	0.544	0.38	-2.268	0.516	0.21
8	-1.887	0.533	0.442	-2.299	0.583	0.485
9	-2.036	0.498	0.345	-2.076	0.57	0.549
10	-1.95	0.536	0.438	-2.107	0.577	0.567
11	-1.874	0.552	0.476	-2.207	0.558	0.409
12	-1.869	0.559	0.521	-2.271	0.573	0.524
13	-1.835	0.565	0.509	-2.173	0.597	0.559
14	-2.042	0.558	0.494	-2.363	0.584	0.585
15	-2.119	0.558	0.533	-2.377	0.593	0.58
16	-2.33	0.57	0.542	-2.624	0.594	0.573
17	-2.209	0.563	0.49	-2.469	0.587	0.517
18	-2.041	0.558	0.532	-2.544	0.576	0.378
19	-2.239	0.578	0.558	-2.338	0.584	0.612
20	-2.212	0.589	0.584	-2.334	0.546	0.244
21	-2.244	0.589	0.596	-2.368	0.586	0.489
22	-2.23	0.599	0.671	-2.583	0.587	0.566
23	-2.39	0.601	0.678	-2.467	0.598	0.552
24	-2.532	0.584	0.467	-2.66	0.589	0.564
25	-2.626	0.556	0.119	-2.378	0.587	0.583
26	-2.656	0.584	0.384	-2.895	0.59	0.473
27	-2.233	0.572	0.483	-2.592	0.59	0.45
28	-2.469	0.6	0.558	-2.36	0.581	0.496
29	-2.63	0.581	0.416	-2.918	0.577	0.405
30	-2.414	0.591	0.391	-2.31	0.576	0.44
31	-2.78	0.577	0.376	-2.511	0.574	0.508
32	-2.664	0.592	0.432	-2.725	0.567	0.364
33	-2.489	0.562	0.347	-2.565	0.594	0.544
34	-2.697	0.577	0.179	-3.026	0.58	0.325
35	-2.478	0.598	0.647	-2.731	0.57	0.286
36	-2.736	0.574	0.379	-3.196	0.557	0.19
37	-2.464	0.588	0.461	-2.84	0.565	0.168
38	-2.987	0.553	-0.11	-2.972	0.564	0.236
39	-3.067	0.571	0.06	-2.948	0.567	0.266
40	-3.056	0.573	0.324	-3.027	0.561	0.104

Appendix F

Sample of Survey for human judgment task (word intrusion)

The highlighted words are just for you to see how difficult it is to identify the intruders

	Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword 5	Keyword 6	Keyword 7	Keyword 8	Keyword 9	Keyword 10	intruder
Topic 1	party	eu	labour	mps	election	leave	vote	protest	deal	tory	
Topic 2	company	beach	passenger	airport	crash	air	animal	fly	plane	flight	
Topic 3	government	force	donald_trump	cathedral	military	war	china	country	president	leader	
Topic 4	visit	police	baby	meghan	family	harry	wedding	prince	birth	queen	
Topic 5	public	woman	claim	investigation	law	court	report	legal	feel	charge	

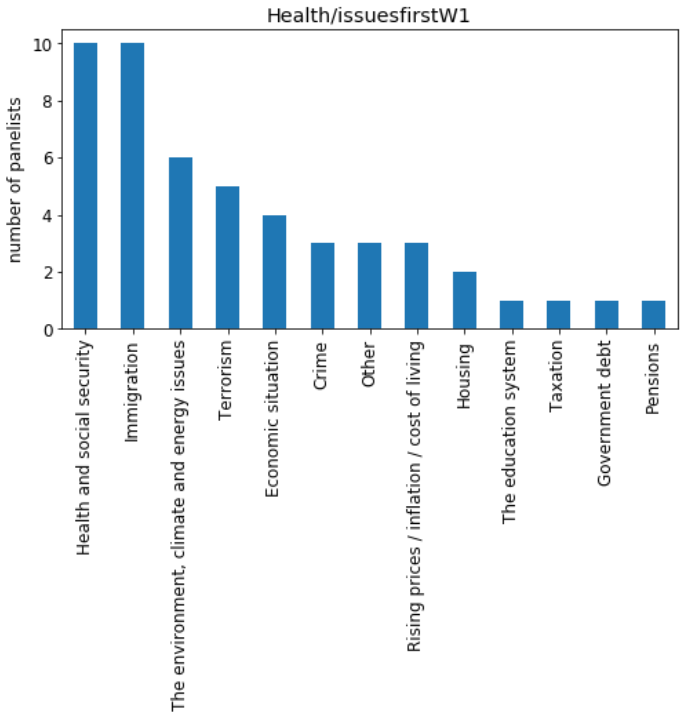
UK evaluation sample

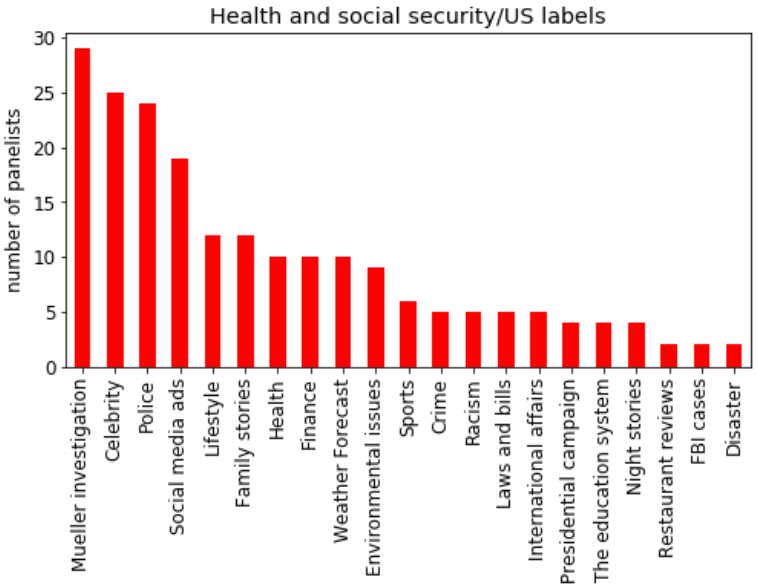
	Keyword 1	Keyword 2	Keyword 3	Keyword 4	Keyword 5	Keyword 6	Keyword 7	Keyword 8	Keyword 9	Keyword 10	intruder
Topic 9	president	house	report	mueller	special_counsel	donald	investigation	congress	food	white	
Topic 10	mother	son	live	parent	love	father	woman	trump	family	life	
Topic 11	local	live	owner	place	house	law	property	restaurant	resident	california	
Topic 12	animal	island	grow	large	climate_change	water	plant	human	police	scientist	
Topic 13	area	park	crash	plane	hour	air	flight	border	car	road	

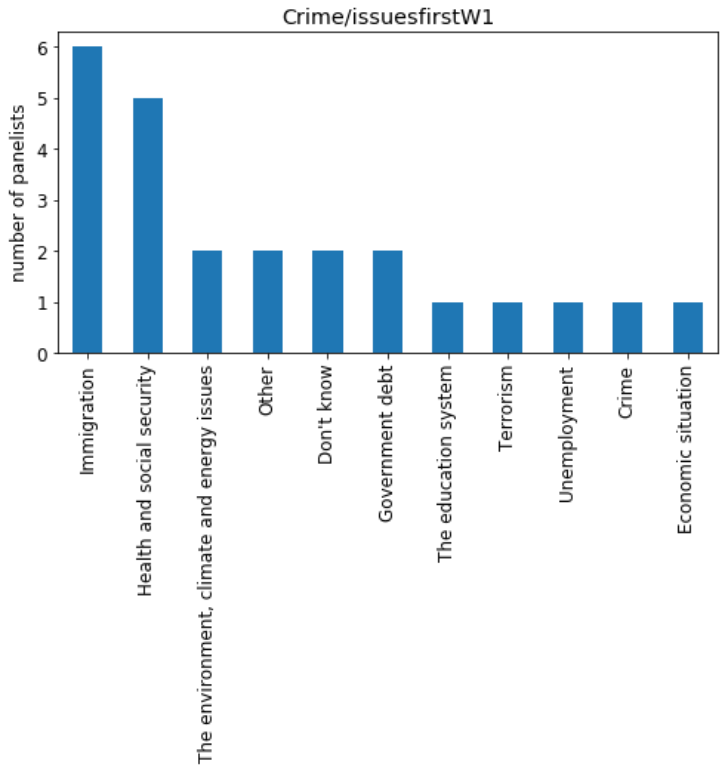
US evaluation sample

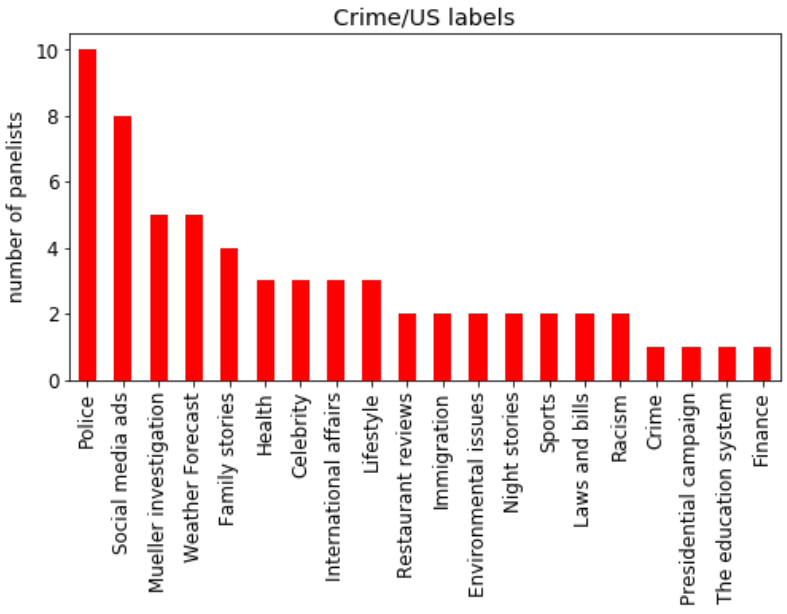
Appendix G

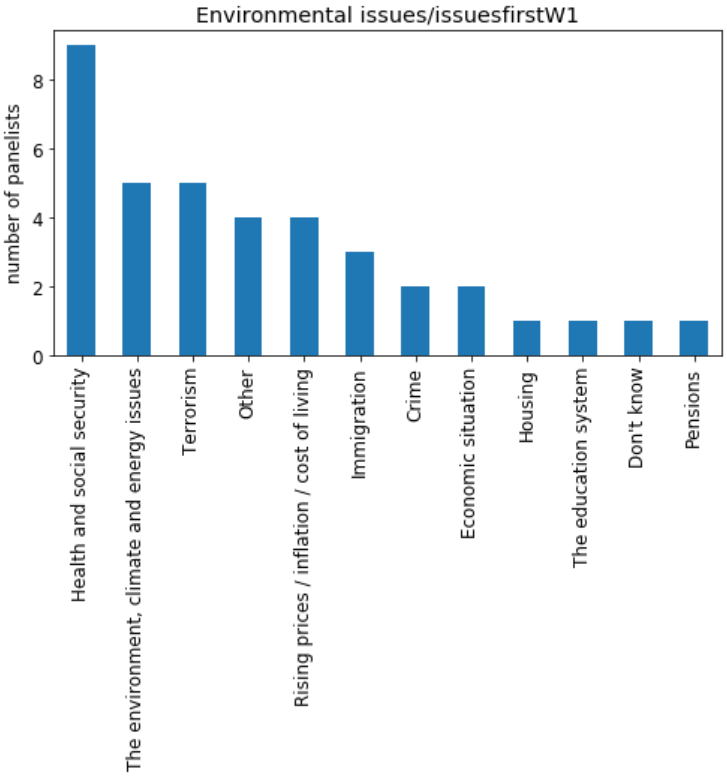
Plots of Survey opinions from the user profile vector when aggregating users based on Top topics for US data in blue.
Plots of Top topics from the user profile vector when aggregating users based on survey opinion in red

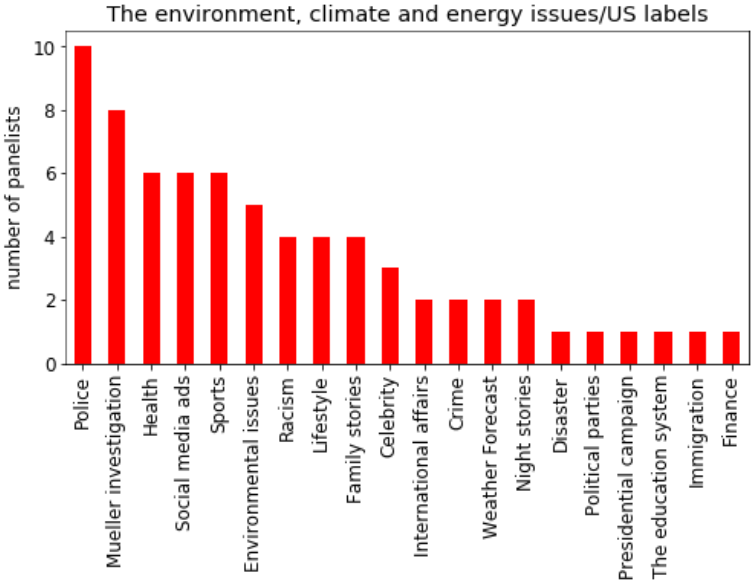


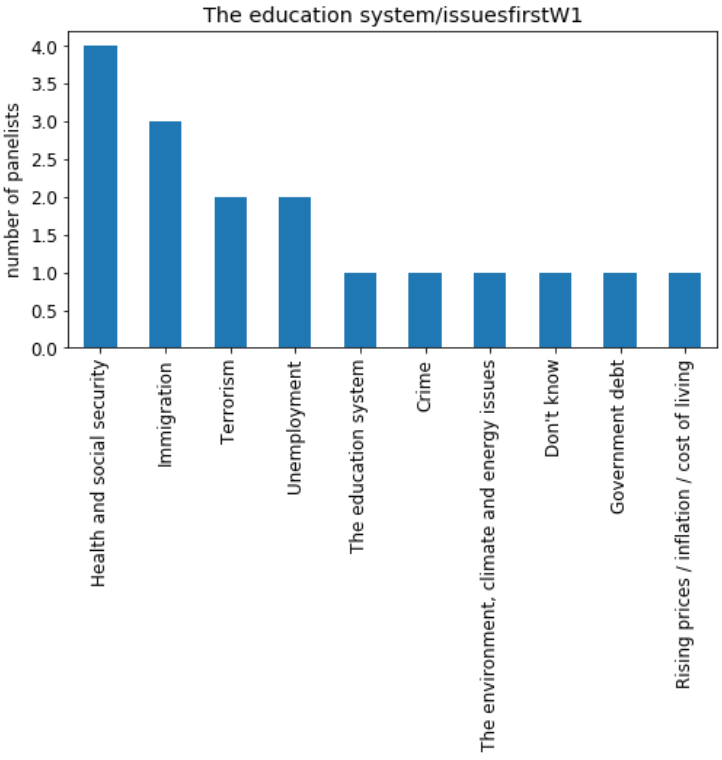


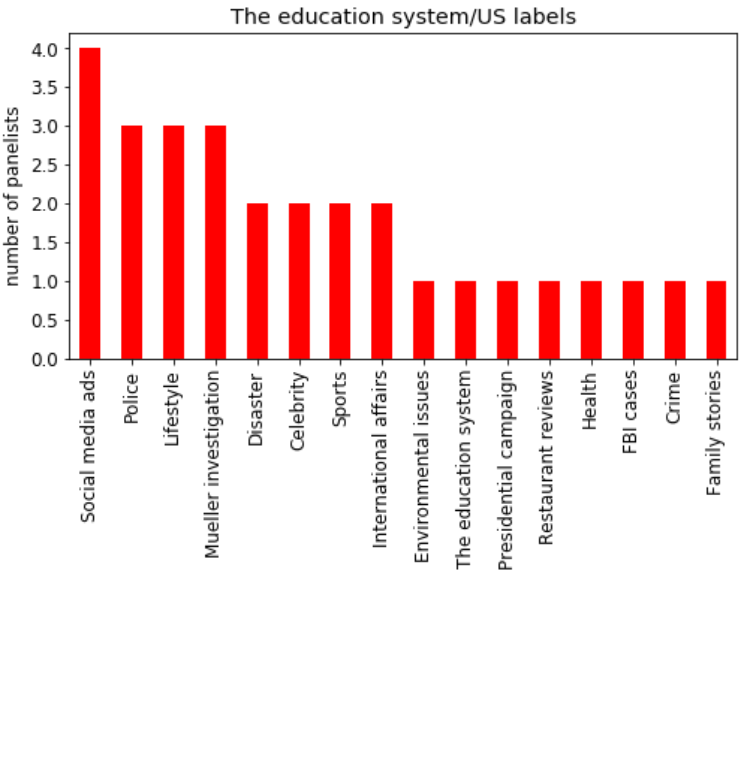




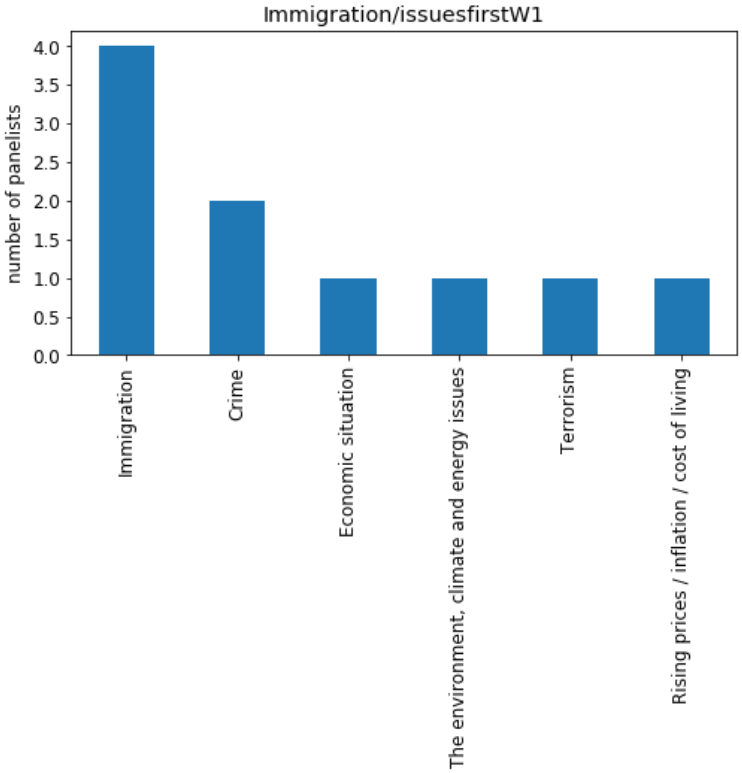

Environmental issues/issuesfirstW1
number of panelists
0
2
4
6
8
Health and social security
The environment, climate and energy issues
Terrorism
Other
Rising prices / inflation / cost of living
Immigration
Crime
Economic situation
Housing
The education system
Don't know
Pensions

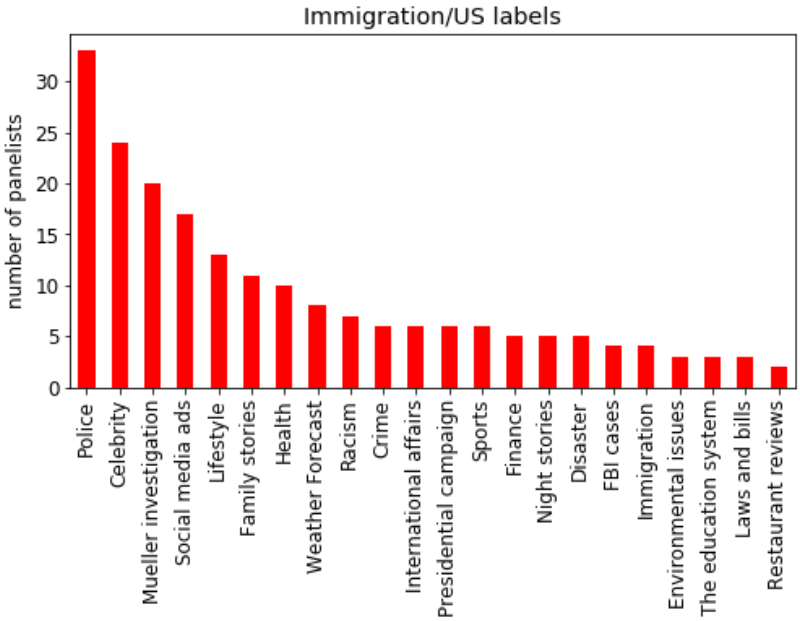

The environment, climate and energy issues/US labels
number of panelists
0
2
4
6
8
10
Police
Mueller investigation
Health
Social media ads
Sports
Environmental issues
Racism
Lifestyle
Family stories
Celebrity
International affairs
Crime
Weather Forecast
Night stories
Disaster
Political parties
Presidential campaign
The education system
Immigration
Finance


The education system/issuesfirstW1
number of panelists
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Health and social security
Immigration
Terrorism
Unemployment
The education system
Crime
The environment, climate and energy issues
Don't know
Government debt
Rising prices / inflation / cost of living


The education system/US labels
number of panelists
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Social media ads
Police
Lifestyle
Mueller investigation
Disaster
Celebrity
Sports
International affairs
Environmental issues
The education system
Presidential campaign
Restaurant reviews
Health
FBI cases
Crime
Family stories

Appendix G (Continued)

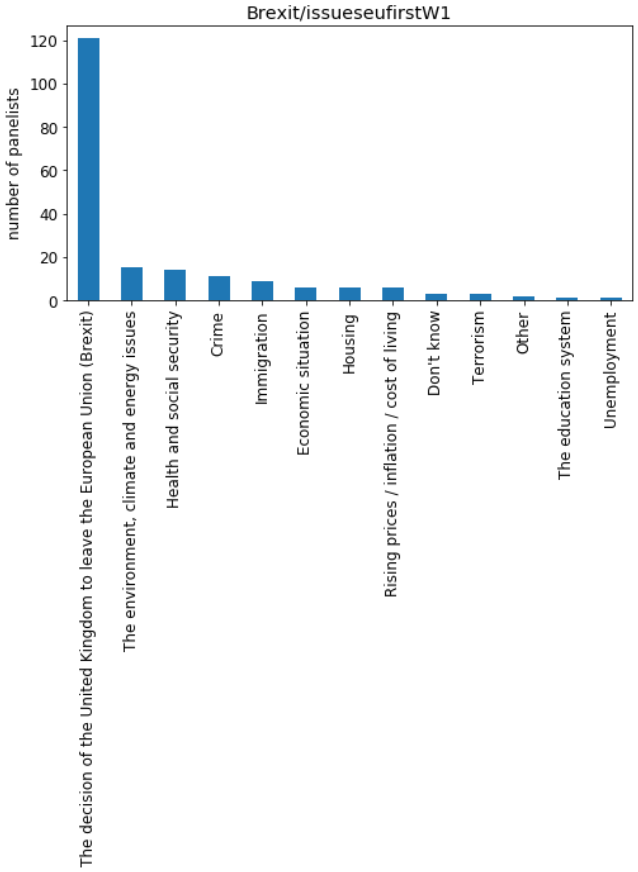


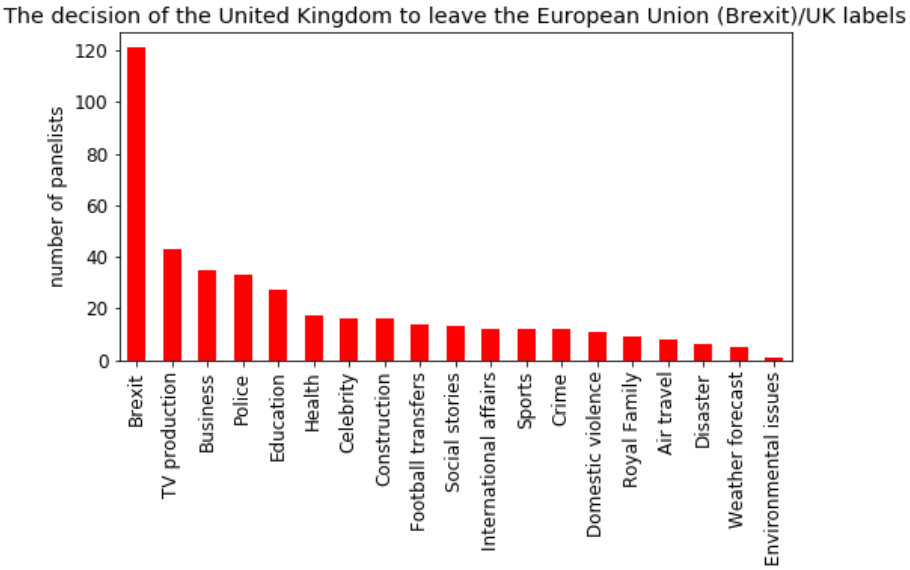


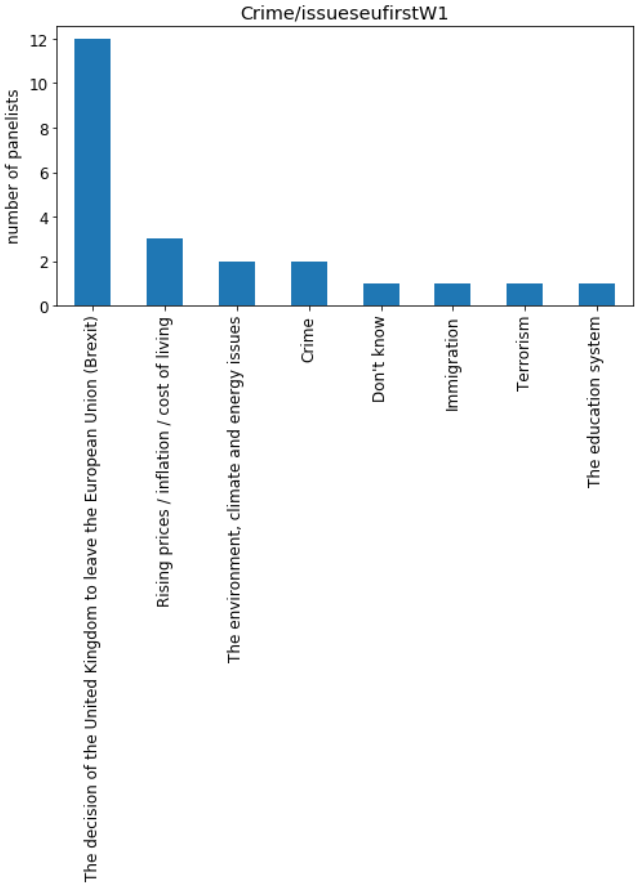
Appendix G (Continued)

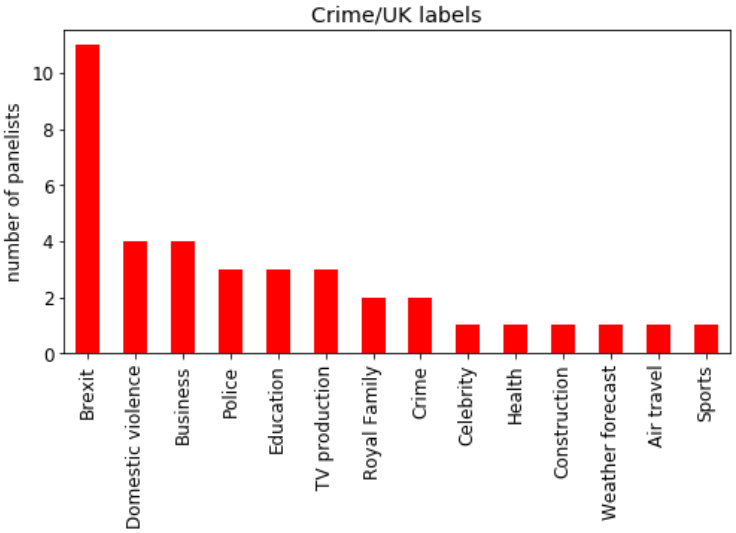
Plots of Survey opinions from the user profile vector when aggregating users based on Top topics for UK data in blue.

Plots of Top topics from the user profile vector when aggregating users based on survey opinion in red.

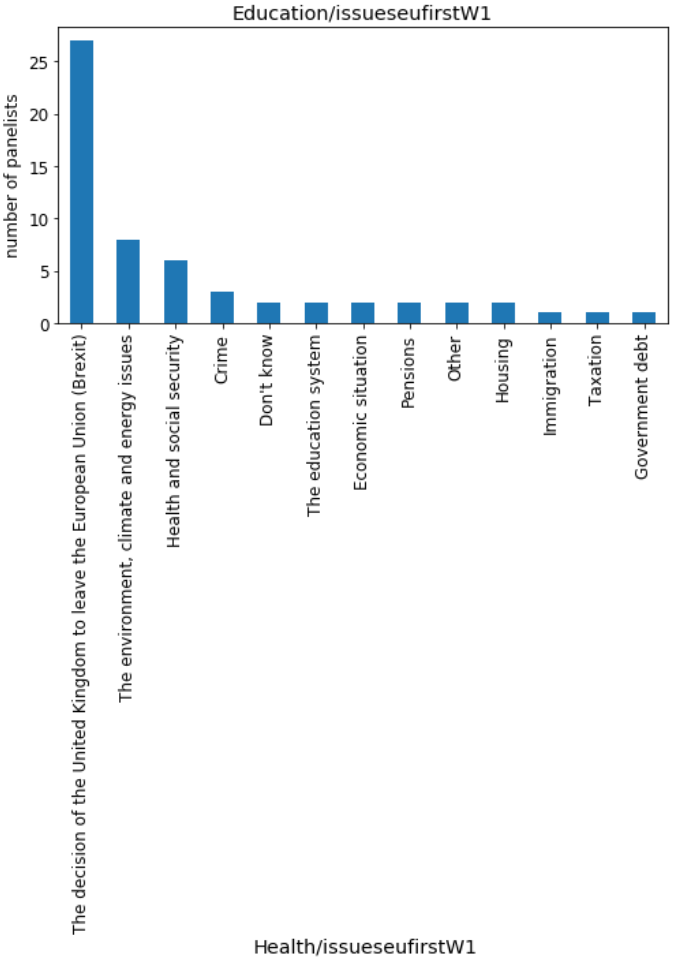


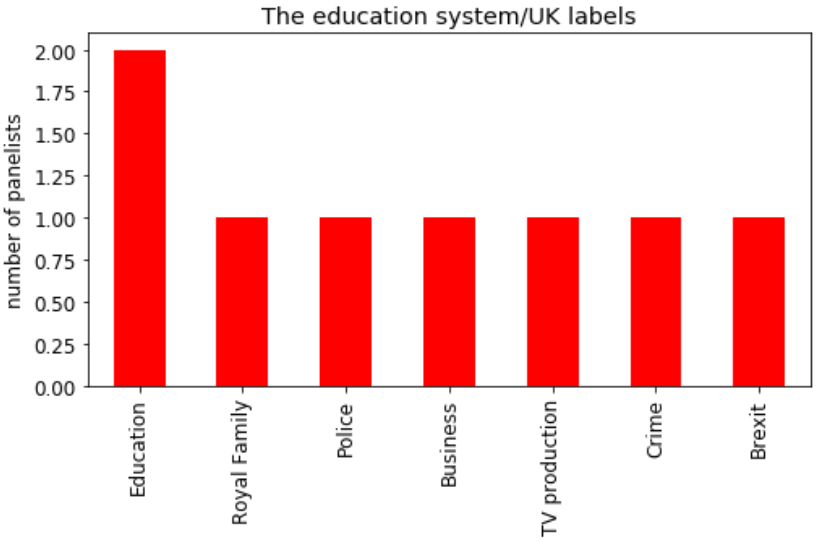


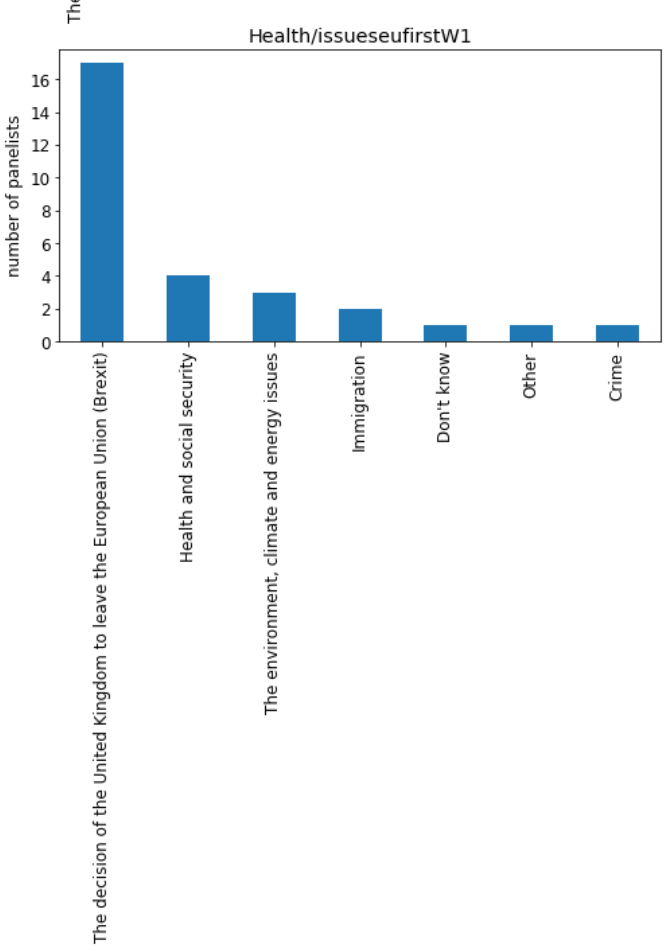


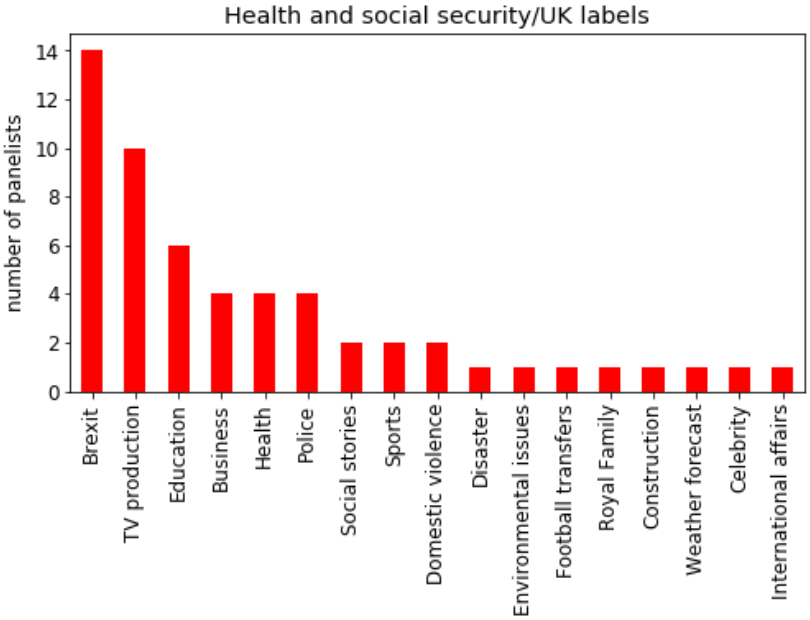


Appendix G (Continued)

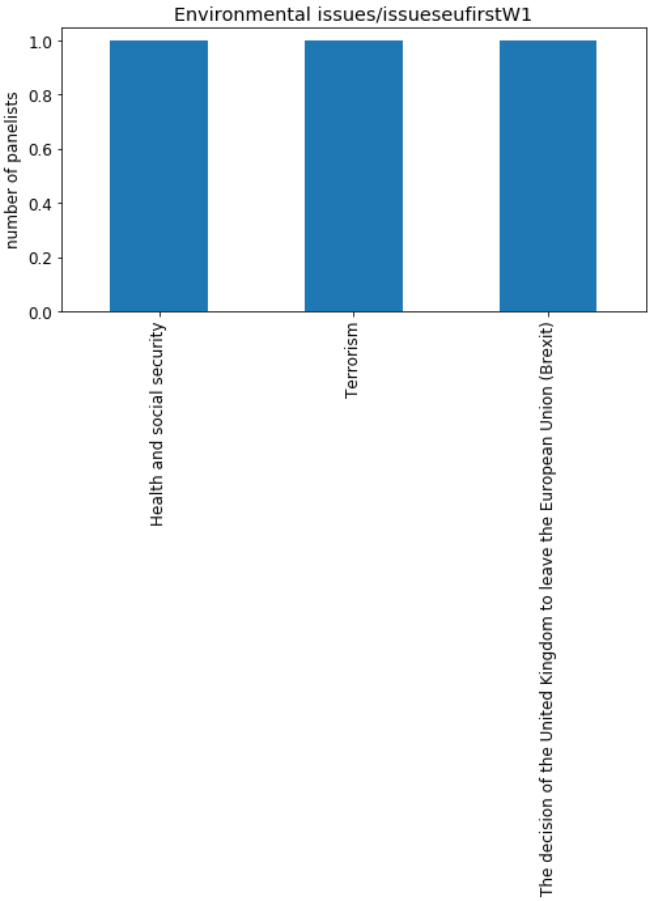


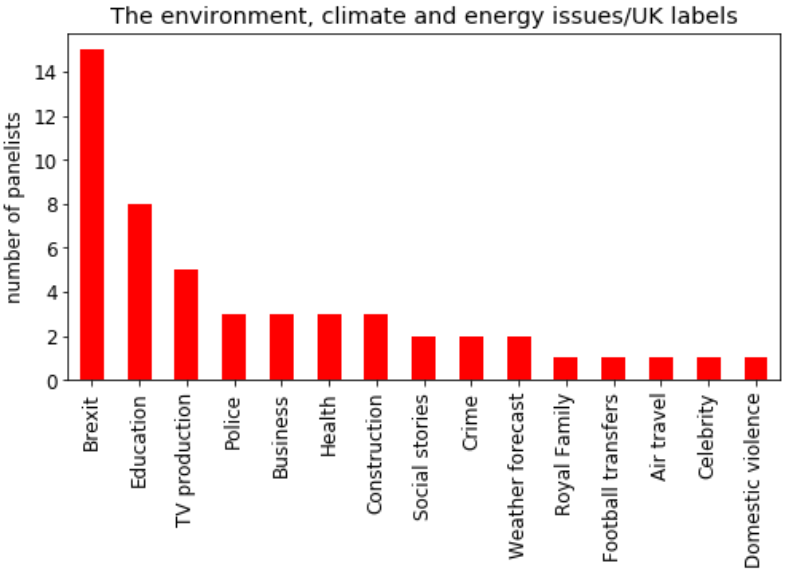






Appendix G (Continued)





References

Anjie Fang, Philip Habel, Iadh Ounis, & Craig MacDonald. (2019). *Votes on Twitter: Assessing Candidate Preferences and Topics of Discussion During the 2016 U.S. Presidential Election.*

Aritz Bilbao-jayo, & Aitor Almeida. (2018). *Political discourse classification in social networks using context sensitive convolutional neural networks.*

David M. Beli . (2012). *Probabilistic topic models.*

David M. Blei, Andrew Y. Ng, & Michael I. Jordan. (2003). *Latent dirichlet allocation.*

Haewoon Kwak, Jisun An, Joni Salminen, Soon-Gyo Jung, & Bernard J. Jansen. (2018). *What We Read, What We Search: Media Attention and Public Attention Among 193 Countries.*

Identification of Topics and Their Evolution in Management Science. (2018). Hentet fra www.lizlance.ca.

Ivan P. Yamshchikov, & Sharwin Rezagholi. (2018). *Elephants, Donkeys, and Colonel Blotto.*

Jonathan Chang , & Jordan Boyd-Graber. (2009). *Reading Tea Leaves: How Humans Interpret Topic Models.*

Marina Sokolova, Kanyi Huang, Stan Matwin, Joshua Ramisch, Vera Sazonova, Renee Black, . . . Nanjira Sambuli. (2016). *Topic Modelling and Event Identification from Twitter Textual Data.*

Michael Röder, Andreas Both, & Alexander Hinneburg. (2015). *Exploring the Space of Topic Coherence Measures.*

Neuman, W. R., Guggenheim, L., Jang, M. J., & Bae, S. Y. (2014). *The Dynamics of Public Attention: Agenda-Setting Theory Meets Big Data.*

Social Politics: Agenda Setting and Political Communication on Social Media. (u.d.).

Wen Zou, Weizhong Zhao, James J. Chen, & Roger Perkins. (2017). *Best Setting of Model Parameters in Applying Topic Modeling on Textual Documents.*

Xinxin Yang, Bo-Chiuan Chen, Mrinmoy Maity, & Emilio Ferrara. (2016). *Social Politics: Agenda Setting and Political Communication on Social Media.*

Yeojin Kim, Chris J. Vargo, William J Gonzenbach, & Youngju Kim. (2016). *First and Second Levels of Intermedia Agenda Setting: Political Advertising, Newspapers, and Twitter During the 2012 U.S. Presidential Election.*

Yue Lu, Qiaozhu Mei, & ChengXiang Zhai. (2010). *Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA.*

Yue Lu, Quaozhu Mei , & ChengXiang Zhai. (2010). *Investigating task performance of probabilistic topic.*