UNIVERSITY OF TRENTO

Department of Sociology and Social Research

Master's Degree
in
Data Science

Analyzing Polarization in Climate Change Tweets during COP: A Multi-Layer Networks and Topic Modeling Approach

Alessio Gandelli

Supervisor: Elena Pavan[1]

Co-Supervisor. Matteo Magnani[2]

Academic year 2022/2023

Acknowledgements

And I would like to acknowledge ...

Abstract

This is where you write your abstract ...

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Background

Climate change has been a well-known problem among scientists for a long time. The first paper warning the effects of the increment of carbon dioxide in the atmosphere is from 1976 [1] while in a study of 1976 [2] they observed it for the first time. In 1988, the World Meteorological Organization Established the Intergovernmental Panel On Climate Change (IPCC) [3], which is up to date as the leading organization evaluating climate change. As shown in [4], the researcher's interest in climate change grew significantly after 1990.

"All the News That's Fit to Print"

The New York Times

VOL.CXXXVII . . . No. 47,546 NEW YORK, FRIDAY, JUNE 24, 1988

Global Warming Has Begun, Expert Tells Senate

Global Warming: Greenhouse Effect?

Average global temperatures through the first five months of 1988. As a baseline, scientists use the global average from 1950 to 1980.

Source: James E. Hansen and Sergej Lebedeff

Sharp Cut in Burning of Fossil Fuels Is Urged to Battle Shift in Climate

By PHILIP SHABECOFF

WASHINGTON, June 23 — The earth has been warmer in the first five months of this year than in any comparable period since measurements began 130 years ago, and the higher temperatures can now be attributed to a long-expected global warming trend linked to pollution, a space agency scientist reported today.

Until now, scientists have been cautious about attributing rising global temperatures of recent years to the predicted global warming caused by pollutants in the atmosphere, known as the "greenhouse effect." But today Dr. James E. Hansen of the National Aeronautics and Space Administration told a Congressional committee that it was 99 percent certain that the warming trend was not a natural variation but was caused by a buildup of carbon dioxide and other artificial gases in the atmosphere.

An Impact Lasting Centuries

Dr. Hansen, a leading expert on climate change, said in an interview that there was no "magic number" that showed when the greenhouse effect was actually starting to cause changes in climate and weather. But he added, "It is time to stop waffling so much and say that the evidence is pretty strong

Drought Raising Food Prices; Inflation Effect Seems Minor

By ROBERT D. HERSHEY Jr.

Cañon Zapata in Tijuana, Mexico, the busiest illegal crossing point.

Fig. 1.1 Front page of NY Times, June 24, 1988

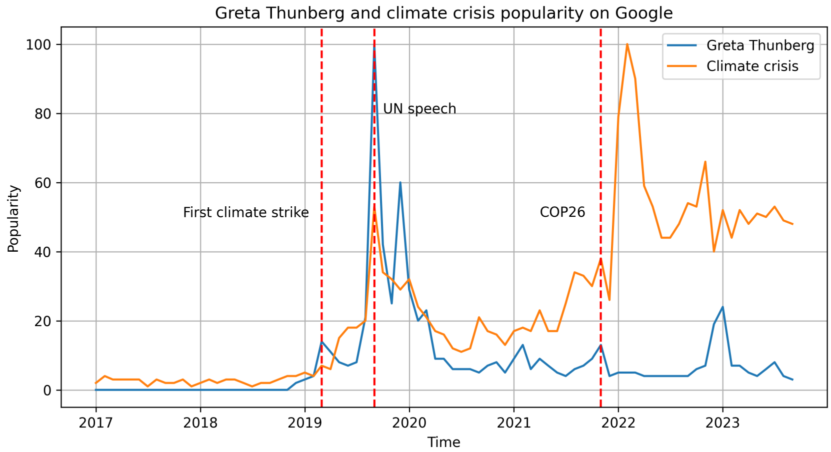

Fig. 1.2 interest on Google over the time of Greta Thunberg and climate crisis

Only in the past few years has it become mainstream, in part thanks to Greta Thunberg [5] and Fridays for Future, people were more aware of the problem. As we can see in Fig 1.2, only after 2019 people started searching (and talking) about a 'climate crisis' [1], underlining the urgency with which we should act. It is also true that the strikes caused inconvenience to many normal citizens trying to go to their workplaces. This means that someone may have developed a bad feeling toward this activism and, thus, toward climate change. An increase in polarization has been detected after the strikes [6].

In particular, Twitter is a place where political debate takes place [7] and political events like Conferences of Parties (COP) are the perfect opportunity to study climate change discussion because they are the conferences where the highest political figures of many countries meet and talk about climate emergencies. The increasing polarization poses a significant challenge to mitigating the harmful impacts of climate change; for this reason, understanting the root cause has the potential to contribute significantly to safeguarding the world from the impacts of climate change.

Conference of Parties Conferences of Parties are yearly conferences organized by the United Nations where the topic of discussion is climate change; the first was held in 1995 in Berlin, and the ones that ended with a document to ratify are:

- **COP3**: Kyoto Protocol (1997) It was the first treaty to mandate countries to cut greenhouse gas emissions legally, but this was true only for developed countries, excluding China and India.

[1] motivo

- **COP21**: The Paris Agreement (2015) is a global accord that aims to limit global warming to well below 2 degrees Celsius, preferably to 1.5 degrees Celsius, compared to pre-industrial levels by requiring all countries to set their own emissions reduction targets. It is considered more effective and inclusive than the Kyoto Protocol because it involves all countries, allows for flexibility in setting emissions targets, and includes a robust system for transparency and accountability.

- **COP26**: Glasgow Climate Pact (2021) is a crucial agreement in the global effort to combat climate change. It includes significant commitments to address the urgent challenges of climate change, such as phasing down coal usage, increasing climate finance for adaptation, strengthening international cooperation, and supporting countries in transitioning to low-carbon economies. However, not everyone agrees with the outcomes of the conference. [8] [9] [10]

Unfortunately, we do not have Twitter data for the first one, but for the latter two, we have, and our focus will be on those two.

This work lays its foundations on the research of Falkenberg et al. [11], where they discovered that cop26 was way more polarized than cop21. Using a similar approach, we will explore the ideological polarization topic by topic.

There is not a universally agreed definition of polarization. In this paper, we will use the one stated in [12], the same used by Falkenberg, which is: "The most common measure of polarization in the political literature is probably bimodality, which is the idea that the population can be usefully broken down into two subpopulations". In our case, the two sub-populations are pro-climate and climate skeptics.

1.2 Research Questions

Thanks to the structure we gave to our research, we can now answer a new set of questions related to intra-topic polarization. The first and most straightforward is RQ1, which aims to inspect the topics that are driving the polarization of the entire COP26. Consequently, RQ2 wants to identify whether these topics have always been polarized.

Then, we will move to some questions related to the users; in particular, RQ3 wants to see if the polarized users are polarized in the same way over the different topics or if there are topics in which they are on the opposite side of the spectrum. RQ4 instead investigates whether the users who talk about many different topics are more polarized than those who are present in only one.

1. Which are the most polarizing topics discussed on Twitter during Cop 26?

2. How did topics evolve between cop21 and cop26?

3. Is the single-user polarization different across different topics?

4. Are the users present in more topics polarized in different ways than the ones that are present only in one or few topics?

Chapter 2

Related Work

In this Chapter we will see the literature about the main topics of this thesis, understanding why studying the climate is important, what is polarization and its connection with the climate debate and the state of the art of topic modeling.

2.1 Topics

Topic modeling is a widely used technique to extract and analyze latent topics from a collection of documents. In this paper, we focus on topic modeling for short text since we are handling tweets, which is challenging due to the limited amount of text available for each document.

In this chapter, we review traditional methods such as LDA and NMF and more advanced techniques like BERTopic, top2Vec, BTM, and GSDMM.

2.1.1 Traditional methods

LDA (Latent Dirichlet Allocation, 2003)[13] is a traditional probabilistic algorithm for topic modeling. It assumes that each document is a mixture of topics, and each topic is a distribution over words. However, LDA struggles with short texts due to the sparsity of the word co-occurrence matrix.

Another classical algorithm based on linear algebra is NMF (Non-negative Matrix Factorization) [14][15]. It assumes that each document is a linear combination of topics, and each topic is a non-negative linear combination of words. It can be used for short texts and as a baseline method.

2.1.2 Advanced methods

Here, we will see the most recent techniques that have been used and tested in different settings:

BTM (Biterm Topic modeling, 2013) [16] unlike the traditional methods, it does not use the bag of words approach, but it considers the co-occurrence of words to the topic modeling. A generative probabilistic model, similar to LDA, represents each document as a set of couples of words (biterms).

GSDMM (Gibbs Sampling algorithm for the Dirichlet Multinomial Mixture model, 2014) [17] is a clustering-based algorithm that uses Gibbs sampling to iteratively assign each document to a topic. It models the distribution of words in each topic using a Dirichlet distribution. GSDMM is designed explicitly for short text from social media, which is typically unstructured and contains slang.

2.1.3 Neural approach

The most advanced techniques (BERTopic [18]and Top2Vec [19]) exploit neural networks, in particular transformers [20], and have a common structure:

1. Vectorization of the document using embeddings
2. Dimensionality reduction
3. Clustering
4. Topic definition

Let us see in detail all these steps:

Embeddings

The first and most crucial step is mapping each document into a vector of numbers in order to be easily manipulated by the machine. We can do this in a wide range of different ways: from naive one-hot encoding (Recurrent Neural Networks (RNN), Long Short-Term Memory Networks) to more complex transformers. Egger et al. [21] proved that the latter is the best-performing method.

A transformer [20] is a deep learning architecture that excels at processing sequential data, such as natural language text. Unlike previous architectures like recurrent neural networks (RNNs) and long short-term memory (LSTM) models, transformers can process all parts of the input data in parallel, which allows for faster training and execution.

The self-attention mechanism works by assigning different weights to different parts of the input. For each token in a sequence, it calculates a score (the attention score) that determines how much focus to place on other tokens when encoding that token. The higher the score, the more focus the model places on the token. An interesting feature of the transformer is the ability to map the same word from different contexts into two different points in a multidimensional space. Different words with the same meaning will be mapped in two close points.

Usually, transformers work at a word level. However, now we are interested in a sentence level, so we will use sentence transformers, which map every document, instead of every word, to a vector.

The goal is to map two documents about the same topic to two multidimensional points that are close according to some notion of distance, usually Euclidean or cosine similarity.

BERTopic traditionally uses a BERT transformer to calculate its embeddings, but the flexibility of the framework allows the use of other embedding methods. The more known are Doc2Vec, Universal Sentence Encoder, and recently Openai released their version of embedder [1].

Dimensionality reduction

Dimensionality reduction is the process of reducing the number of input variables or features while retaining the essential information and preserving the underlying structure of the data. After embedding the documents, we have a set of long vectors; for instance using the *all-MiniLM-L6-v2* model, you get a 384 vector long for each document. Since our goal is to cluster the documents, this algorithm works better with low dimensional data due to the well-known curse of dimensionality problem [22]. For this purpose, there exist many algorithms that can reduce the length of the vectors. Usually, for clustering, less than ten dimensions are suggested. Some examples are: Principal Component Analysis [23](PCA), which works maximizing the variance of the data along the principal components, t-SNE [24]focuses on preserving the local structure of the data in the lower-dimensional space. However, t-SNE is not recommended for clustering or outlier detection as it does not necessarily preserve distances or densities well, UMAP [25], which is particularly known for its scalability, ability to preserve global structure, and computational efficiency compared to other dimensionality reduction methods. These features make UMAP suited to reduce the dimensionality of big datasets like the one used in this research.

[1] https://platform.openai.com/docs/guides/embeddings/what-are-embeddings

Clustering

Clustering is an unsupervised machine-learning technique that aims to group similar objects together based on their features. The goal of clustering is to partition the data points into subsets, called clusters, such that objects within the same cluster are related to each other in a certain way. There are plenty of clustering algorithms: hierarchical, density-based, distribution-based, centroid based.

K-means clustering [26] is an iterative algorithm that partitions a dataset into a predefined number of clusters based on the proximity of data points to cluster centroids.

HDBSCAN [27](Hierarchical Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm that extends the DBSCAN algorithm to handle noise and clusters of different densities. It can automatically determine the optimal number of clusters and allow outliers.

These properties of HDBSCAN make it perfect for clustering tweets since they are noisy, and not all tweets should be put in a cluster.

Topic Definition

At this point, we are able to give a numeric label to each document depending on the cluster; the next is to give each cluster a set of representative words. Here we have the difference between Bertopic and Top2Vec,

Bertopic uses Term Frequency- Inverse Document Frequency (TF-IDF) [28] statistic to find the most relevant word for each topic; let us see in detail how it works:

Term Frequency: It's the ratio of the number of times a specific term appears in a document to the total number of terms in that document.

Inverse Document Frequency: The IDF of a specific term can be calculated as the logarithm of the total number of documents in the corpus divided by the number of documents containing the specific term. This gives more importance to rare words in the corpus(the collection of documents)

TF-IDF is the product of these 2.

$$TFIDF(t) = TF(t) \cdot IDF(t) \tag{2.1}$$

Top2Vec, instead of each cluster, first identifies the centroid, which is typically calculated as the mean value of every dimension of all the points in the cluster. Then, get the n closest words to the centroid.

Topic Representation

After extracting the most meaningful words of each topic, it is useful to give a label, which is a small sentence that can distillate in the best way the content of the cluster. This is not an easy task, and it has traditionally been performed by humans, consuming time and energy and not getting the best result possible since humans are biased by their opinions [29] [30]. Using Large Language Models, in particular, GPT gives a new opportunity to perform this task with good accuracy quickly. We will see in the methods chapter how it performed in this scenario.

2.1.4 Topic Evaluation

An important step was to evaluate the performance of these models, either by using unsupervised metrics or by setting up a custom experiment using a labeled dataset. The first method uses metrics well known in the field of topic modeling and will be presented here, while the second part can be found in the chapter on methods.

Metrics The coherence and diversity metrics are measures used in topic modeling to evaluate the quality of the learned topics. Coherence is a measure of how often words co-occur, more than would be expected if they were independent. Diversity measure the difference between two topics in term of words used.

- **NPMI (Normalized Pointwise Mutual Information)** The NPMI value lies between -1 and 1, where a higher value indicates a higher level of coherence, thus implying better topics. It is calculated as follows: For every pair of unique words, $(\text{word1}, \text{word2})$, in a given topic, the Pointwise Mutual Information (PMI) is calculated as:

$$\text{PMI}(\text{word1}, \text{word2}) = \log\left(\frac{P(\text{word1}, \text{word2})}{P(\text{word1}) \cdot P(\text{word2})}\right) \tag{2.2}$$

  Where $P(\text{word1}, \text{word2})$ is the probability of both words appearing in a sliding window, and $P(\text{word1})$ and $P(\text{word2})$ are the individual probabilities of each word appearing in a window.

  PMI values are then normalized to ensure the measure is not overly influenced by the frequency of word pairs:

$$\text{NPMI}(\text{word1}, \text{word2}) = \frac{\text{PMI}(\text{word1}, \text{word2})}{-\log(P(\text{word1}, \text{word2}))} \tag{2.3}$$

The NPMI coherence of a topic is the average of the NPMI values of all pairs of words in that topic. The NPMI value lies between -1 and +1, with a higher value indicating better topic coherence.

- **Umass coherence**: similar to npmi, for each pair of words, $(\text{word1}, \text{word2})$, the UMass coherence score is calculated as:

$$\text{UMass}(\text{word1}, \text{word2}) = \log\left(\frac{\text{co\_count}(\text{word1}, \text{word2}) + \varepsilon}{\text{count}(\text{word1})}\right) \tag{2.4}$$

Where $\text{co\_count}(\text{word1}, \text{word2})$ is the number of documents in which the two words co-occur, $\text{count}(\text{word1})$ is the number of documents in which the first word appears, and ε is used to avoid logarithm of zero.

The UMass coherence of a topic is then the sum of these log values for all pairs of words in the topic. The UMass coherence score can take on any value from negative infinity to zero, with values closer to zero indicating higher coherence.

- **diversity**: In the context of topic modeling, the diversity score is a measure that quantifies how different the topics are from each other. One way to compute this is by looking at the proportion of unique words across all topics.

$$\text{Diversity Score} = \frac{\text{Number of Unique Words}}{\text{Total Number of Words}} \tag{2.5}$$

- **Computation time**: time needed to fit the models

2.2 Networks

Social Network Analysis (SNA): Social Network Analysis is a field that examines the relationships, interactions, and structures within a network of individuals or entities. It provides valuable insights into the dynamics, information flow, and influence within social networks. Several studies have applied SNA in different domains, such as online social networks [31], organizational networks, and public health networks. In this scenario, we are using it to explore the topology of the interaction between users of Twitter; we are studying the structure of their interaction to see if there are some recurring patterns.

In particular, we will use Multi-layer networks [32]: complex networks that capture multiple types of relationships or interactions between nodes. They allow us to represent different dimensions or contexts in a single framework, providing a more comprehensive understanding of network dynamics. Thanks to the complexity added by the multiple layers, we can see how the same users interact in different dimensions, in our case, topics. This implies that for each identified topic in the topic modeling phase, we can observe a network of interactions among users, allowing us to study the presence of the users on multiple topics.

2.3 Polarization

As anticipated in chapter 1, this work make sense only after understanding what Falkenberg did in his research [11]. Since we are following the same methods, we also do the same assumption, such as the bipolarity of the polarization. In this section we will see how polarization will be computed.

A meaningful metric that gained popularity among social scientists is polarization. The researchers believe that polarization can be harmful to mantain the democratic stability [33]. Thus understanding the phenomenon is important to develop a solution to irìt.

It has been used to study the impact of political discussion on social media, especially around US presidential elections [34] [35] . Due to this, we should be careful generalizing since US politics is built around two main parties (Democrats and Republicans), so a bimodal view of polarization is the best suited for this case, but not for all. Despite this, an analysis of the polarization over 21 different countries shows that the US is not the only place where it has been detected [36].

Even though Falkenberg detected an increasing polarization only in the COP26, in 2021, Williams et al. [37] found the presence of echo chambers around the climate discussion in social media, with a small presence of open forums. In this work, we will try to connect these two pieces of research to understand if, breaking down the discussion into topics, we can see if the polarization of specific topics has always been high.

There is not a clear and universally adopted definition of polarization; Bramson et al. [12] tried to summarise it defining different types: Spread, Dispersion, coverage, Regionaliation, Community fractioning, distinctness, group coverage, group consensus, and size parity.

We use the Dispersion polarization definition that looks into the distribution of beliefs to detect peaks. Falkenberg demonstrated that the assumption of bimodality makes sense, dividing the population into climate supporters and climate skeptics. We also assume that we can detect the polarization using the retweet network.

Latent ideology In order to compute polarization on a retweet network, we first estimate a latent ideology for each user, as defined in [38] and adapted for retweets in [35].

Starting from the adjacency matrix of the retweet network and after some linear algebra, we can obtain a latent ideology score for each user.

The first step is, out of the n users, to identify m most retweeted users, we will call them influencers, then build an adjacency matrix $A \in \mathbb{R}^{nxm}$ between users and influencers (where a_{ij} is the number of times user i retweeted influencer j). Now let us see in detail the process from the matrix to the scores. It this way we are excluding from the analysis all the users that did not interact with with the top n influencers.

First, normalize A by the number of retweets:

$$P = \frac{A_{ij}}{\sum_i \sum_j a_{ij}} \tag{2.6}$$

Next, get the vector of row and column sums and consider the diagonal matrix:

$$\mathbf{r} \in \mathbb{R}^m, \quad r_i = \sum_i a_{ij} \tag{2.7}$$

$$\mathbf{c} \in \mathbb{R}^n, \quad c_j = \sum_j a_{ij} \tag{2.8}$$

$$R = \begin{bmatrix} \frac{1}{\sqrt{r_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{r_n}} \end{bmatrix} \quad C = \begin{bmatrix} \frac{1}{\sqrt{c_1}} & & \\ & \ddots & \\ & & \frac{1}{\sqrt{c_n}} \end{bmatrix} \tag{2.9}$$

Then, compute the matrix of standardized residuals S:

$$S = R\left(P - (\mathbf{r} \cdot \mathbf{c}^T)\right)C \tag{2.10}$$

Using Singular Value Decomposition (SVD), which is a factorization technique in linear algebra, we can decompose the standardized matrix into three other matrices. It provides essential geometrical and theoretical insights about linear transformations and is extensively used in various fields such as data science, engineering, and statistics [39]. Given matrix S, its SVD is written as:

$$S = U\Sigma V^T \tag{2.11}$$

where U is an $m \times m$ matrix whose columns are the orthonormal eigenvectors of AA^T, Σ is an $m \times n$ diagonal matrix whose non-zero elements are the singular values of A, and V^T is the transpose of an $n \times n$ matrix whose columns are the orthonormal eigenvectors of A^TA. The singular values on the diagonal of Σ are typically sorted in descending order.

The columns of U and V are called the left-singular vectors and right-singular vectors of A, respectively.

Multiply R and U:

$$X = RU \tag{2.12}$$

Finally, rescale U on $[-1, 1]$ and get the user score:

$$score = -1 + 2 \cdot \frac{X - \min(X)}{\max(X) - \min(X)} \tag{2.13}$$

Hartigan's diptest After computing all the users' latent ideology scores, to test the polarization, we use Hartigan's diptest [40].

Hartigan's Dip Test is a statistical test used to determine if a distribution is unimodal. The test works by comparing the empirical distribution function of the data, denoted as $F(x)$, to the unimodal distribution function that minimizes the maximum difference between $F(x)$ and itself, denoted as $G(x)$. The dip statistic D is then defined as:

$$D = \sup_x |F(x) - G(x)| \tag{2.14}$$

Where $\sup_x$ denotes the supremum (least upper bound) overall x, the unimodal distribution function $G(x)$ is chosen such that it minimizes this supremum. In other words, $G(x)$ is the "best" unimodal approximation to the empirical distribution function $F(x)$.

The null hypothesis of the Dip Test is that the data comes from an unimodal distribution. If the dip statistic D is significantly large, we reject the null hypothesis and conclude that the data is not unimodal. The p-value of the test is computed by comparing the observed dip statistic to the distribution of the dip statistic under the null hypothesis. This distribution is typically approximated using Monte Carlo simulations.

Chapter 3

Data Description

In this chapter, we will see an overview of the starting data used in this research, as well as some general statistics about it.

3.1 Where the data comes from

The data are tweets collected from the Twitter API containing the hashtag #cop21 #cop26. For each cop, we have two jsonlines files, one for the tweets and one for the users involved. The fields are the ones stated in the documentation [1], there are many but the relevant ones to us are the following:

Field	**Description**
author	The ID of the author
author_name	The username of the author
text	The text of the tweet
date	The creation date of the tweet
lang	The language of the tweet
conversation_id	The ID of the conversation the tweet belongs to
referenced_type	The type of the referenced tweet
referenced_id	The ID of the referenced tweet
mentions_name	The usernames of the mentioned users in the tweet
mentions_id	The IDs of the mentioned users in the tweet

Table 3.1 Description of the fields used of the tweets data

[1] https://developer.twitter.com/en/docs/twitter-api/data-dictionary/object-model/tweet

While we use the user's file to map the ID of the users to their username, even though we do not always have this information, in that case, we use the user ID as the username.

3.2 Some statistics

We have data from 2 cops: COP21, COP26. All the tweets are in English and without links or image/video content. We call an original tweet a tweet written by a user, so that's not a retweet. Fig 3.1 shows the distribution of the tweets over time for both cops; most of the tweets have been tweeted while the conferences were taking place.

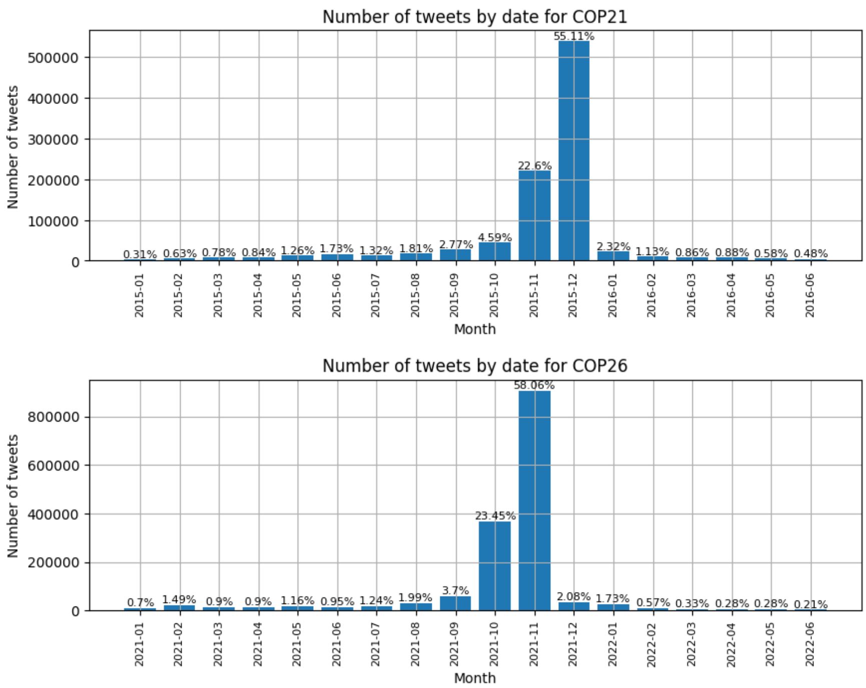

Fig. 3.1 Numebr of tweets by date for cop 21 and cop26

COP21 The tweets span from January 2015 to June 2016, but 77% of the tweets are from November and December 2021; cop26 was held between 30th November and 12th December. In the dataset, 975040 tweets have been tweeted by 234389 users, of which only tweeted an original tweet with at least one retweet; every user tweeted on average 4.16 tweets; the maximum amount of tweets a user tweeted is 9635, 89% of users tweeted less than five tweets.

COP26 The tweets span from January 2021 to July 2022, but 81% of the tweets are from October and November 2021; cop26 was held between 31st October and 12th November. In

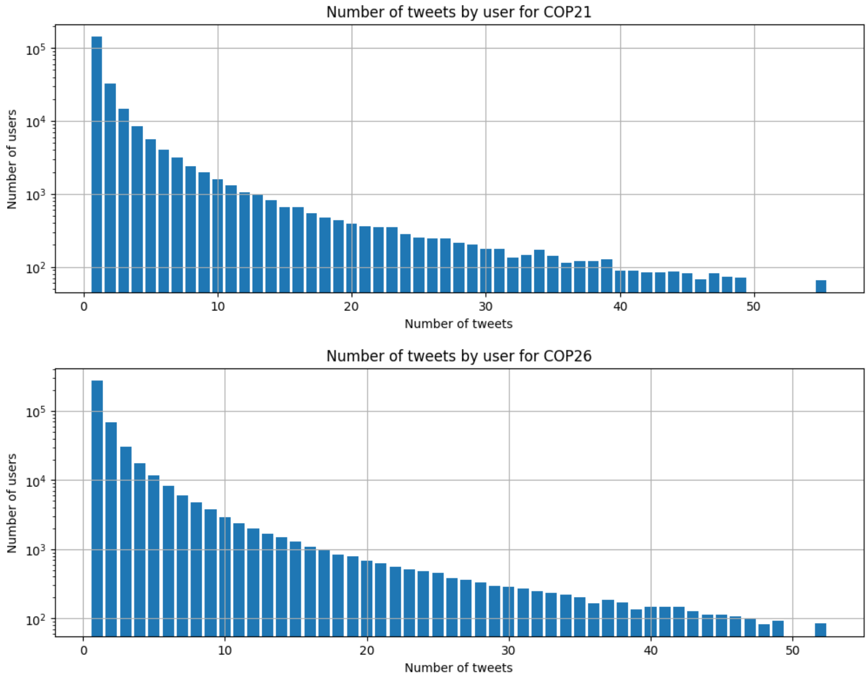

Fig. 3.2 Numebr of tweets by user for cop 21 and cop26

the dataset, 1558968 tweets have been tweeted by 456000 users, of which only 30195 tweeted an original tweet with at least one retweet, every user tweeted on average 3.42 tweets, the maximum amount of tweet a user tweeted is 14267, 90% of users tweeted less than two tweets.

Fig 3.2 shows how most users tweeted just a few tweets (note that it is logarithmic).

	n_tweets	**n_retweets**	**n_original**	**n_original_with_retweets**
COP21	975 040	562 946	412 094	138 427
COP26	1 558 968	1 191 813	367 155	130 138

Table 3.2 Number of tweets

In fig 3.3 we can see how the 1'558'968 are distributed, in fact 76% of them are retweets generated by only 130k original tweets. It is also worth noting that almost 2/3 of the original tweets have 0 retweets.

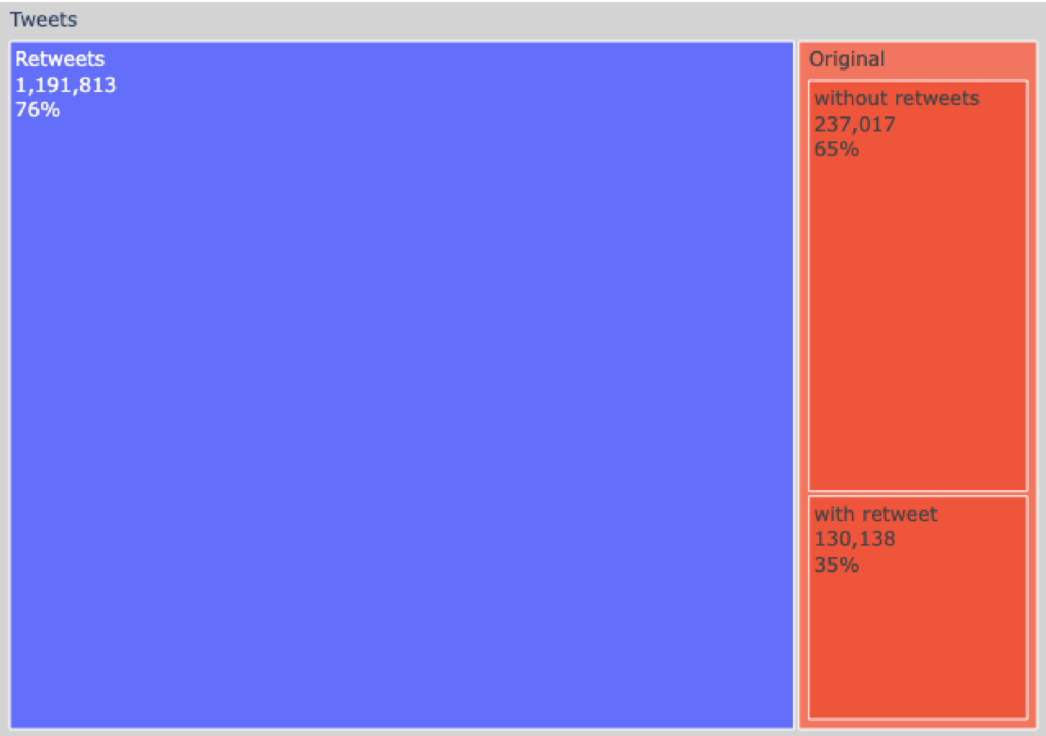

Fig. 3.3 tweets of cop26

Chapter 4

Methodology

In this chapter we will

4.1 Topic Modeling Evaluation

This section presents the evaluation of different models used for tweet labeling. Both unsupervised and supervised approaches were used to evaluate the performance of the models. The evaluation aimed to find the best-performing model to label tweets accurately. The models included traditional methods (LDA, GSDMM, and NMF) and neural models like BERTopic.

The unsupervised evaluation evaluated traditional metrics used in this context, such as coherence and diversity scores. We will see with more details in 4.1.1 The unsupervised evaluation results showed that BERTopic performed better than traditional methods, especially when using all-MiniLM-L6-v2 (BERT) [1] , text-embedding-ada-002 (OpenAI) [2]

and tweet_classification [3] embeddings.

The supervised evaluation consisted of building a custom-labeled dataset from scratch and then looking at the accuracy of the models. The results showed that BERT and OpenAI were the best-performing models. The section concludes with a summary of the results and a description of the representation used for labeling the tweets.

The models used are both traditional(LDA, GSDMM, NMF) as a reference of the ground truth and neural because they seem to be the most accurate; in particular, we will evaluate BERTopic with several embedding methods. We choose BERtopic over Top2Vec because

[1] huggingface.co/sentence-transformers/all-MiniLM-L6-v2

[2] platform.openai.com/docs/models/embeddings

[3] huggingface.co/louisbetsch/tweetclassification-bf-model

they are very similar and also because the Python library is more comprehensive and allows us to be more flexible.

Evaluating a topic modeling algorithm is challenging due to the lack of objectivity in identifying a topic. In this work, we evaluated the models in two ways: first, using a widely used unsupervised approach: metrics like coherence and diversity. Then, to validate the results, we also did a supervised evaluation using different datasets built ad hoc for this setting.

4.1.1 Unsupervised

To compare the different models, we used a library suggested by the creator of BERtopic called OCTIS [41] [42]; this allowed us to structure an experiment to measure different metrics presented in chapter 2: coherence and diversity.

Dataset In this case, the dataset is composed of 1669 preprocessed tweets related to climate change with the hashtag *#cop22*; the preprocessing phase involved removing retweets, links, punctuation, and the most common hashtags (#cop22, #climatechange #p2), all the tweets were in English.

Methods The models used in this evaluation were LDA, NMF, and BERTopic. In the BERtopic case, several embeddings have been tested (all-MiniLM-L6-v2, text-embedding-ada-002, climatebert [43], tweet_classification, USE [44]).

Each model has been fitted several times, changing the parameters:

- **number of topics** from 10 to 50 with a step of 5
- **min topic size**: 5 and 15 tweets[4]

Each unique combination of parameters has been fit three different times; then, we took the mean value of the three computations.

Results The results show that BERtopic performs way better in these tests than the traditional methods. In comparison, the best Bertopic embeddings are mini, OpenAi, and tweet classification.

The experiment demonstrates how the *min_topic_size* value of 5 is too small, so that the results will be with a value of 15.

Fig 4.1 shows the value of all the metrics with a different number of topics for the traditional methods and the best-performing neural one (OpenAI)

[4]only for bertopic

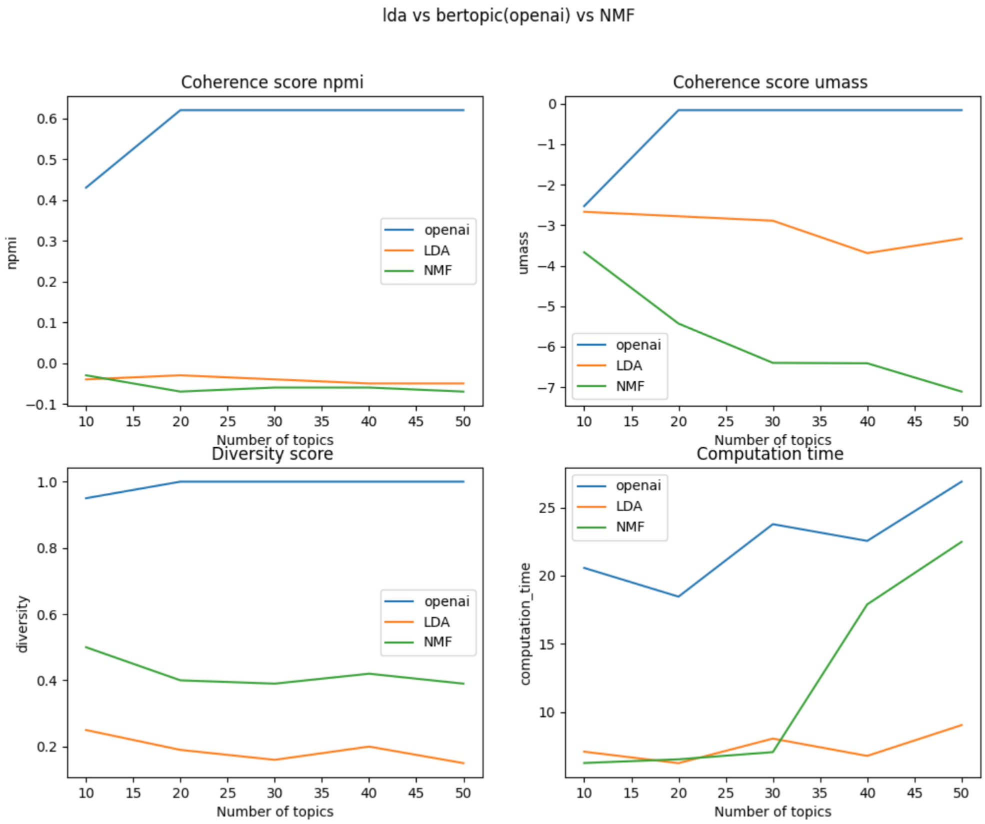

Fig. 4.1 coherence and diversity for LDA, BERTopic, and NMF

However, Hoyle et al. [45] showed how these metrics are not very meaningful for evaluating these models, and we should take these results with a grain of salt.

From this evaluation, we can conclude that BERTopic's topic size is better than smaller, especially if we have many documents. Topics of Bertopic are way more diverse than LDA and NMF, and within the topics, the most relevant words are more semantically related.

4.1.2 Supervised

Considering the result of the unsupervised evaluation, we should use another method to validate what we found. In this case, we created two ad-hoc datasets to see how the models perform in a real-case scenario.

Dataset The first step for the supervised part was the data collection. In this case, we packed specific datasets to test our models. The datasets have been chosen based on the trending topic on Twitter in that days (March 2023)

model	npmi	umass	diversity
tweet_classification	0.62	-0.16	1
openai	0.58	-0.63	0.99
climatebert	0.20	-2.28	0.83
U.S.E	0.20	-4.35	0.89
BERT	0.20	-5.46	0.97
LDA	-0.04	-3.07	0.19
NMF	-0.06	-5.80	0.42

Table 4.1 all the models tested in the unsupervised evaluation

The first is simpler and contains very different topics, so it should be easy to cluster the documents. In contrast, the second is trickier because it includes only politics-related tweets, including some overlapping with more hashtags related to US politics.

- **simple**: 1093 labeled tweets of 5 different topics identified by a hashtag [5]
- **politics**: 1492 labeled tweets of 7 politics-related hashtags [6]

For both datasets, we used two different versions: with and without hashtags. The reason for this is to avoid the model to cluster based on the hashtags.

The tweets have been extracted using twarc2, getting only English tweets and without retweets.

Metrics In order to evaluate the topics, we had to define some metrics. Using BERTopic, we do not set a number of topics apriori, so it has to figure it out by itself. After running the model, we have both the known topic (the hashtag) and the inferred one (a number), and we can now create a confusion matrix between the two sets. To map the inferred topic to the known one, we use the inferred topic with the highest value. This is not always true, but by combining this value with other metrics, we can detect the error.

The metrics defined are the following:

- **Accuracy**: first for each known topic, look at the biggest of inferred topics and divide by the number of tweets in that topic.
- **accuracy no outliers**: in the Bertopic case, the label -1 refers to outliers. Compute the same as accuracy but not counting the outliers

[5] #Bitcoin, #stormydaniels, #UkraineRussianWar, #SaudiArabianGP, #climatechange

[6] #IndictArrestAndConvictTrump, #kabul, #BidenHarris2024, #KamalaHarris, #taiwan, #belarus, # stormydaniels

- **Min_topic_share**: same as accuracy but in the opposite direction, after having computed it for all of my_topics, we take the minimum. This is helpful to detect when the accuracy is considered the wrong topic. This could happen when the inferred topic is less than the actual topic, so one inferred topic contains tweets from multiple topics, and then this number is low.

Parameters max_df is used to remove the terms that appear too frequently; a value of 0.95 means remove the terms that appeared in more than 95/min_df is the opposite; in this case, being an integer, it refers to the minimum number of documents a term should be in to be considered *ngram_range* defines the number of consecutive words to be considered; for example, a value of (1,2) tells the method to consider single words and bigrams (two consecutive words)

$alpha$ is a parameter that influences the number of clusters that will be created; low alpha results in many clusters with single words, while high alphas results in fewer clusters with more words.

```
BERTopic: (nr_topics = 'auto', min_topic_size = 50)

NMF: (max_df = 0.95, min_df = 3, ngram_range = (1,2))

GSDMM: (alpha = 0.1, min_df = 0.1, n_iters = 30)
```

Simple Dataset Results We started evaluating the *simple* dataset with hashtags. As we can see in Fig 4.2 base (all-MiniLM-L6-v2) and OpenAi obtained almost a perfect score for each topic. At the same time, climatebert seems to have a great accuracy but a low mean topic share, this is a signal that something is wrong and we should inspect the heatmap.

In fact, we can clearly see in 4.3 that even though the accuracy is very good, climatebert has some difficulties in dividing the topics, putting almost all the tweets in the same inferred topic. While the first two are performing very well as expected, it is not true for the others. We can see how climatebert put almost all the tweets in topic 0, being able only to find the formula1 tweets and not the climatechange one, as it is designed to do. That's the reason why we decided to remove the models that are not performing well in the simplest case, with the exception of NMF, to use it as ground truth.

Fig 4.4 shows how BERT and OpenAI performed in the simple dataset but without the hashtags, in particular, how BERT tends to find more outliers than OpenAI. Overall, both get a good performance.

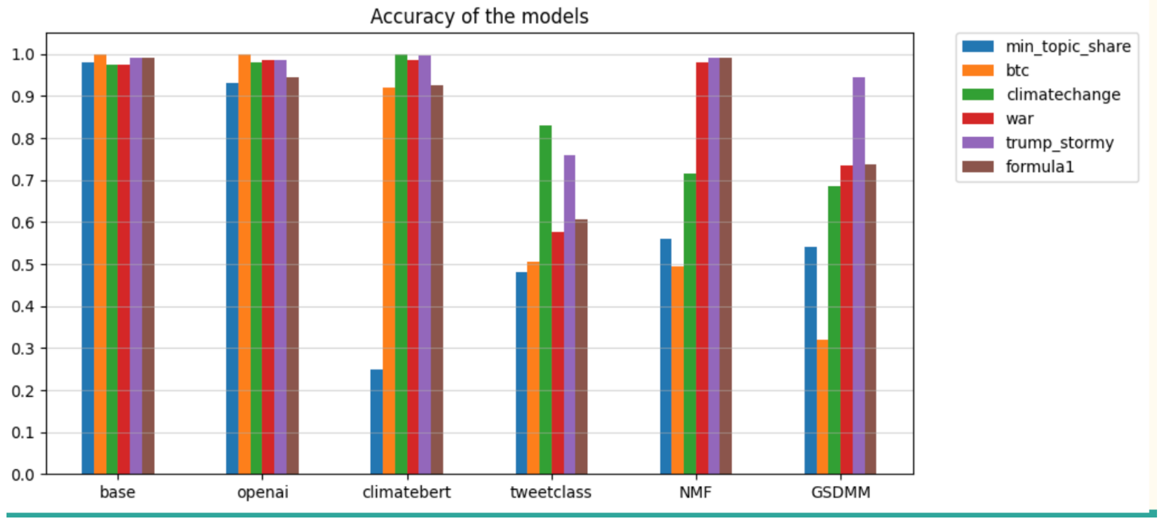

Fig. 4.2 All models accuracy simple with hashtags

An interesting feature of Bertopic is the ability to visualize the different topics in 2-dimensional space; Fig 4.5 shows the document distribution of OpenAi after projecting the embeddings in a two-dimensional space.

Politics dataset results The politics dataset is clearly more difficult to evaluate, but with the hashtags, it is still doing a good job. Fig 4.6 4.7 shows heatmap and topic distribution for the politics dataset with hashtags.

Both BERT and OpenAi are creating two topics from the Taiwan case. OpenAi merges two topics, which makes sense since the two hashtags related to Trump are related to the same event(#trump and #trump_stormy)

To validate the results, we ran the algorithm 100 times, and most of the time, for BERT, the min topic share is 0.9, which means they got the correct number of topics and classified them in a good way.

In the case without hashtags, OpenAi and BERT put in a single cluster all the tweets related to American politics, both also understanding that Kamala's tweets were about something else.

Topic representation The last step is giving a meaningful label to the clusters created; in this manner, we can see if the openai API works well; in particular, we used the model named *gpt-3.5-turbo*. It worked surprisingly well; table 4.2 contains the label generated for both simple and political datasets.

You can use Fig4.6 as a reference. Note the discussion under the #taiwan hashtag is divided into two different topics as present in the document representation.

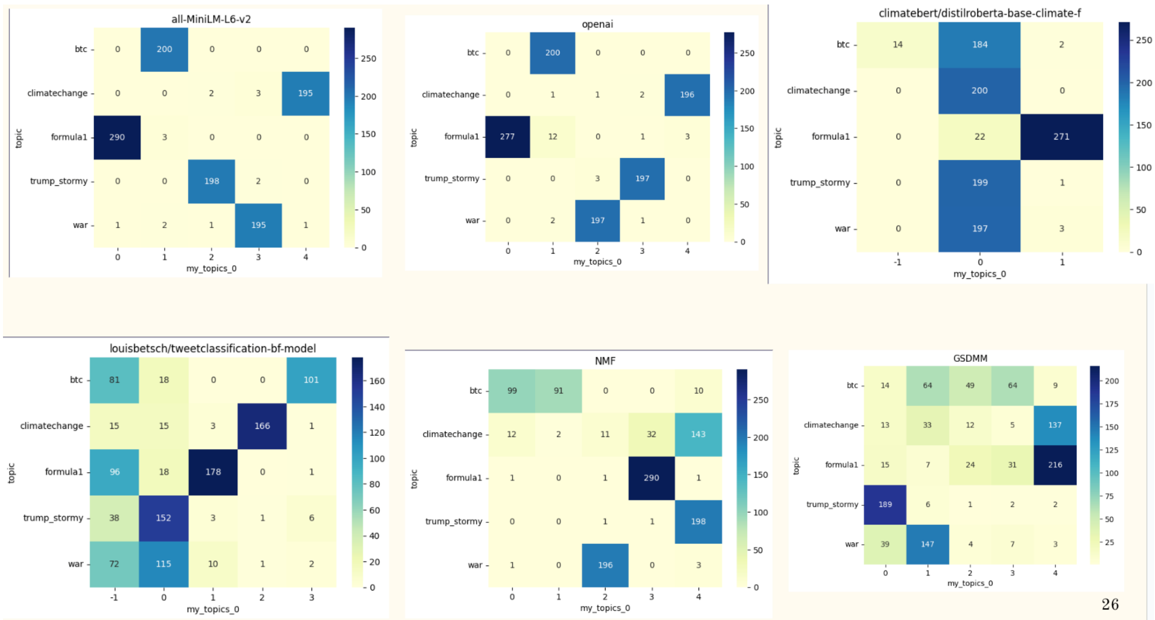

Fig. 4.3 Heatmap comparison of the different model with the simple dataset with hashtags

The prompt used is: *you are a tweet labeler, you are given representative words from a topic and three representative tweets, give more weight to the words, given all this information give a short label for the topic (max 10 words), starts all with topic:*

simple dataset	**label**
#Bitcoin	Cryptocurrencies
#stormydaniels	Trump's hush money payment to Stormy Daniels.
#UkraineRussianWar	Ukraine-Russia conflict
#SaudiArabianGP	F1 Saudi Arabian Grand Prix 2023
#climatechange	Forests and Climate Change
politics dataset (openai)	
#IndictArrestAndConvictTrump	Stormy Daniels controversy
#stormydaniels	Stormy Daniels controversy
#kabul	Suicide bombing near foreign ministry in Kabul
#BidenHarris2024	Politics and Leaders
#KamalaHarris	Kamala Harris official visit to Ghana and Africa
#taiwan	Tensions between China and Taiwan over undersea cables cut
#taiwan	use of small drones for warfare'

Table 4.2 labels generated using GPT API both for simple and political dataset

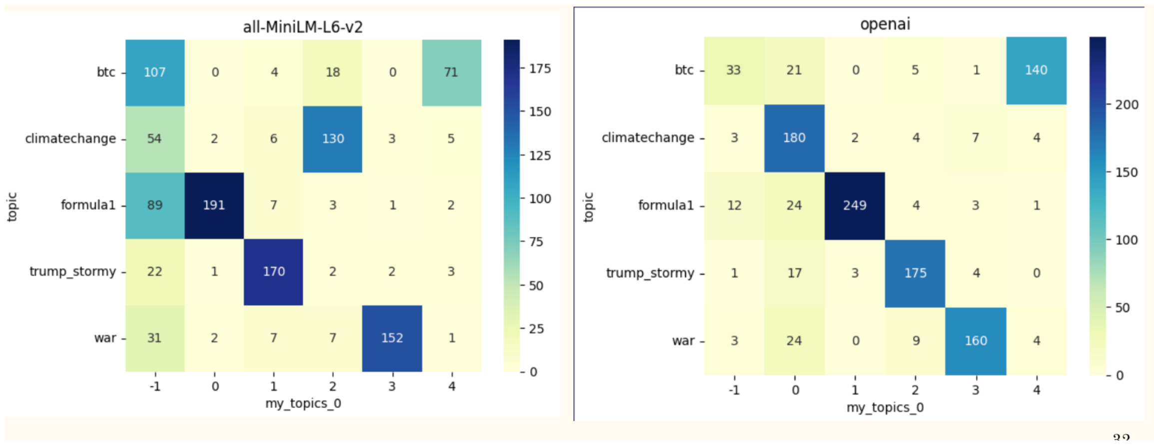

Fig. 4.4 Heatmap comparison of mini and OpenAi of the simple dataset without hashtags

model	accuracy	topic share
BERT	0.84	0.86
OpenAI	0.83	0.85
NMF	0.78	0.69

Table 4.3 recap of supervised evaluation

Conclusion Tab 4.1 shows the result of unsupervised evaluation, while Tab 4.3 shows the supervised. With the supervised evaluation, we showed how the results of the unsupervised one were not completely true; this helped to discard some models and confirm the hypothesis that the neural model with Bert Embedder and Openai was the best performing. There is still a significant difference between the two; Bert is open source and can be run locally, while Openai is not free and can only be used through API.

4.2 Multilayer Network

In this section, we will dive into all the unexplored paths, starting from Falkenberg's work. In particular, we will do the same polarization analysis on a topic level; instead of computing it at a full network level, we created a retweet network for each topic so we can see which are the topics that are driving the polarization of cop. Furthermore, we also want to explore how the polarization of topics evolved over time.

Thanks to the previous section now, we are familiar with the concept of topic modeling and how the main models perform. The goal is to create a multilayer network where a topic represents each layer.

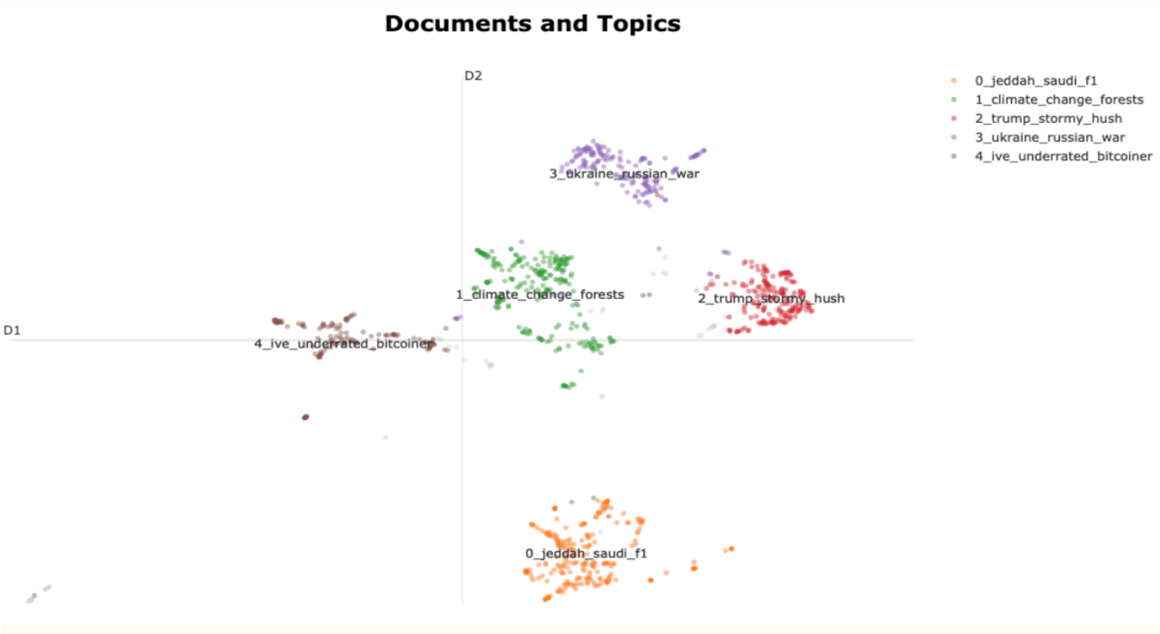

Fig. 4.5 docs representation of simple dataset for openai

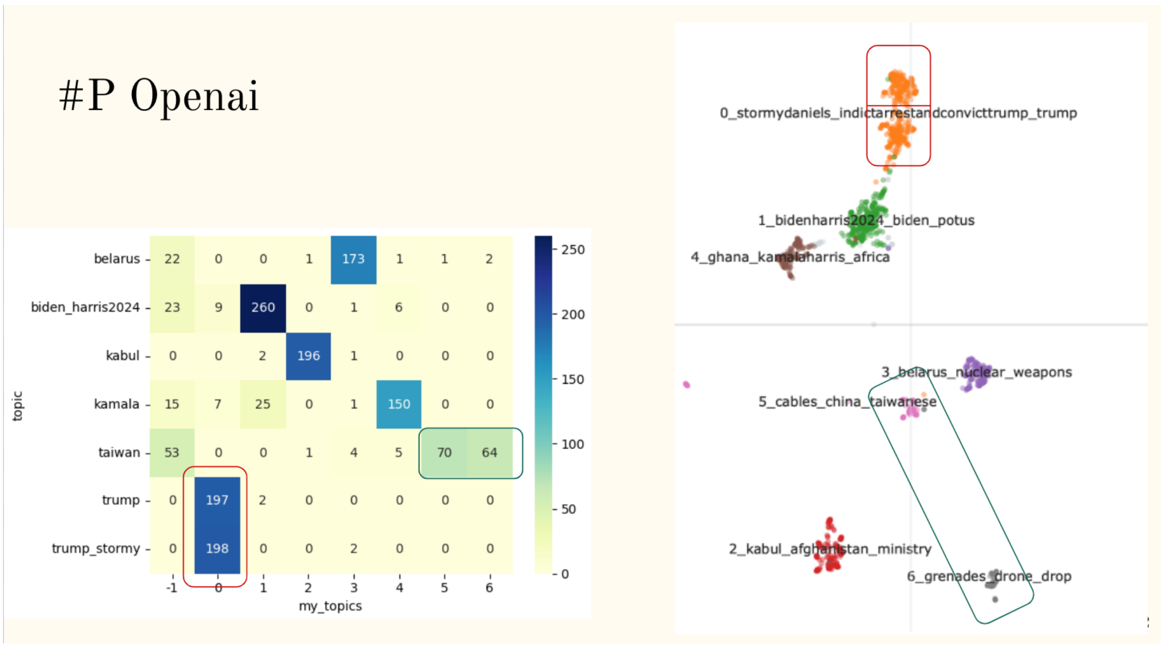

Fig. 4.6 Heatmap and documents representation of the politics dataset with hashtags evaluated with openai

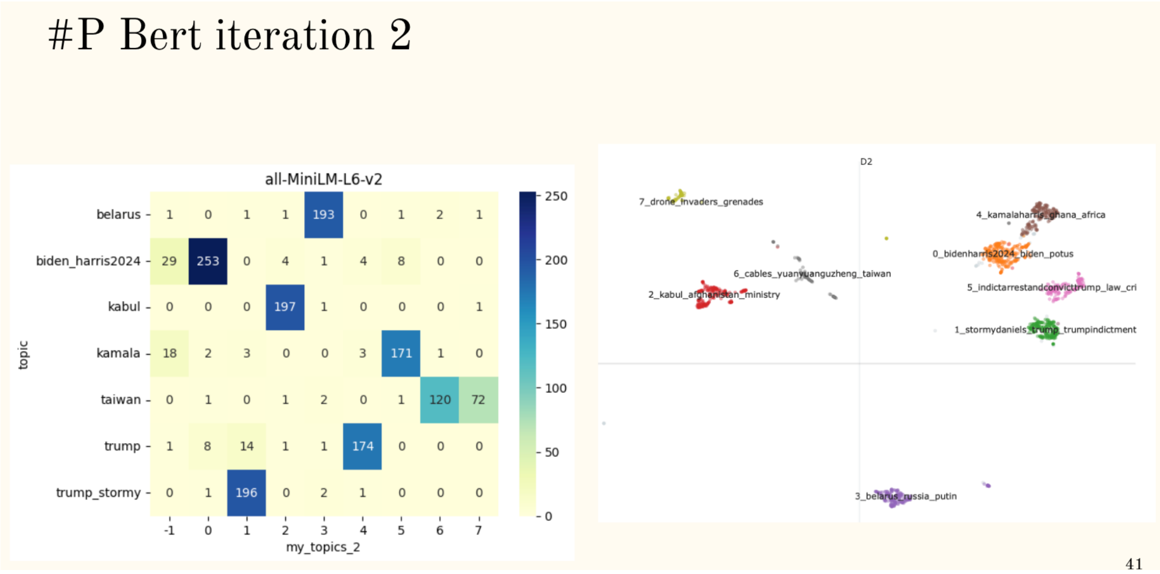

Fig. 4.7 bert heatmap and document viz politics with hashtags

In order to do so, we developed a Python library that can be used as a toolbox starting from the tweets fresh out from the official API of Twitter. The design is modular and can achieve different goals. In fact, even if we are interested only in the retweet network of the users (nodes are users, ties are retweets), this framework can be used to:

- Labeling tweets according to their topic
- Create a temporal text network
- Create the retweet network (normal and multilayer version)
- Create the reply network
- Create the quote network

The steps are independent, so, for example, you can also create the network without the need to run the topic modeling part.

Steps Even though you can skip some steps and start with your own data, the natural and minimal pipeline follows these steps:

1. from JSON to a tabular format
2. label each tweet with a topic
3. create multilayer retweet network

4.2.1 Process input

The first step consists of the transformation of the JSON objects into tabular data to optimize the space and handle the data in an easier way with pandas. This is also helpful to save space; in the case of cop26, we pass from a 14GB JSON to a less than 2 GB CSV since most of the fields are not relevant to this study.

In this process, all the tweets with attachments and not in English are discarded. The tweets are divided into multiple dataframes, one for original tweets, i.e., the ones that the author actively writes, and one for the retweets, quotes, and reply.

At the end of this stage, a CSV and pkl file are saved in case somebody needs the tweets in tabular data. Also, for caching purposes, in fact, if you re-run the script and these files exist, they will be loaded.

4.2.2 Topic modeling

As we extensively discussed in chapter 2 in this segment of the pipeline, the tweets can be labeled using Bertopic, with the possibility to choose the embedder; the one used in this research is *all-MiniLM-L6-v2*.

This step is the most computationally expensive; for this reason, to avoid redundancy, the topic modeling has been run only to original tweets.

After this step, all the original tweets are labeled with a topic, and then the label has been propagated to all the retweets so that the entire dataset is now labeled with a topic.

At this point, it is possible to use the OpenAI API to give a meaningful label to the topics; before this, it was just the most relatively frequent words of the topic. Using the langchain library, we can structure a prompt to be used. This is the one I used:

I want you to act as a tweet labeler, you are given representative words from a topic and three representative tweets, give more attention to the words, all the tweets are related to climate change and COP, there is no need to mention them, detect subtopics. start with "label:" and avoid hashtags, which is a good short label for the topic containing the words [words]?, here are 3 tweets to help you: first = "tweet1", second = "tweet2", third = "tweet3

Similarly to the previous stage, the labeled dataset is saved in the cache folder both in CSV and pkl. The model and the labels are saved, too.

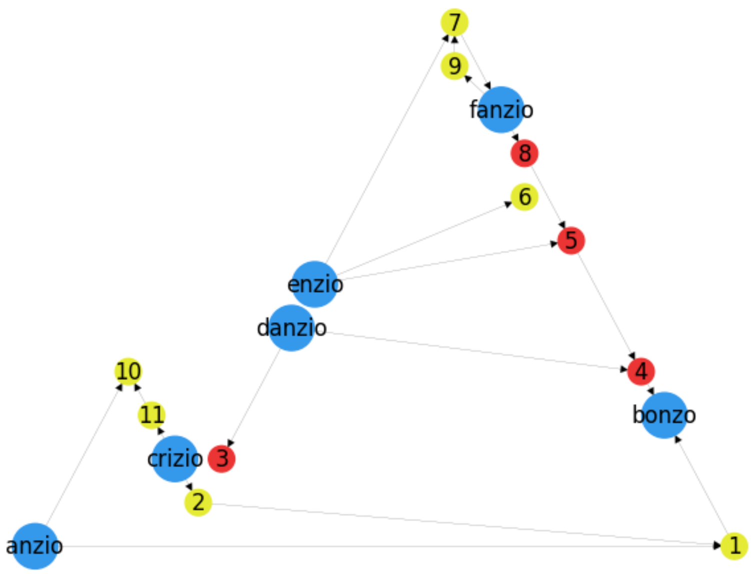

Fig. 4.8 This is a visualization of a temporal text network; the bigger nodes are the users, the smaller are the tweets, and the color is the topic

4.2.3 Temporal text network

should I add this chapter?

This step creates a temporal text network according to Vega and Magnani [46]

The input of this stage is the dataframe containing all the tweets just created, but due to its modularity, you can use your own.

There are two kinds of nodes: users and tweets.

There are 3 kinds of edges :

- **user-tweet**: if the user has tweeted the tweet
- **tweet-user**: if the tweet mention the user
- **tweet-tweet:** if the tweet retweets the other

The method also returns two dictionaries that map tweet nodes to their text and edges to their timestamp; this information is also stored as attributes in the network.

The graph is stored in GML format in the network folder.

4.2.4 Project the network

The temporal text network contains much information and can be used for many different projects, but sometimes, having a simpler data structure is more helpful.

To simplify the analysis, we want to have only one set of nodes, the authors, and see how they are related: this process is known as projection. The network is directed, and the rules for connecting the users are the following:

- if user a is retweeting a tweet of user b: $a \rightarrow b$
- if user c mentions d in a tweet: $c \rightarrow d$

Generally, a tie between two users means that the first did an action towards the second; the action can be a retweet or a mention.

This is achieved using a hybrid approach using both iteration and recursion: first, it iterates over all the users, and for each user iterates on all its tweets, then it recursively searches for the end of the retweet chain.

All the ties take into account the topic of the tweets, so multiple networks are created at this point, one for each topic, all saved in GML format.

4.2.5 Retweet network

While the projection of the temporal text network creates an interaction network, including both retweets and mentions, it is also possible to create a pure retweet network bypassing the temporal text network. The network is directed, the nodes are the users, and the tie is the number of retweets. For each topic, a network is created.

In the process of the creation of the network, there are retweeted tweets that do not have the original one, so we discard them.

4.2.6 Multilayer network

The last step is the creation of a multilayer network using the multinet library developed by Uppsala University; at this point, all the different networks created in the previous step are merged into a multilayer network.

4.3 Polarization

At this point, for each layer, we can compute for each user a latent ideology score, and then, using Hartigan's diptest, we can assign to each topic a polarization value. More details on

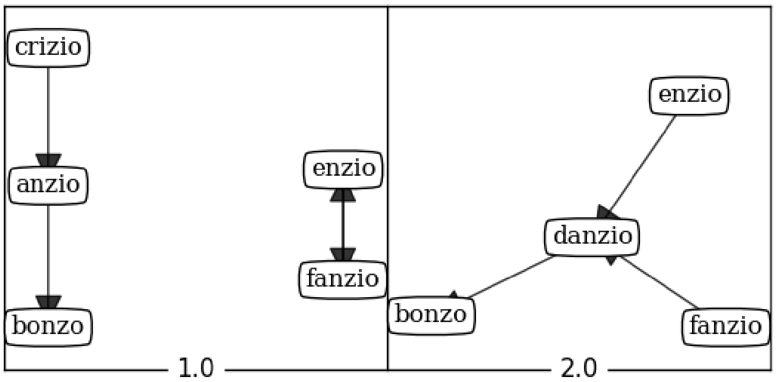

Fig. 4.9 Example of multilayer retweet network 4.8

how this is computed are in the related work. In this process, there are some parameters we can adjust: the number of influencers, which are defined as the most retweeted users, and n, which represents the minimum number of retweets a user should have done to an influencer to be considered.

The ideology score is not computed on all the users, but when selecting the influencers, we are delimiting the users to the ones that retweeted those influencers.

Now we have for those users the ideology score for each of them.

4.4 Logitudinal analysis

In order to see topic polarization over time, we need to run the topic modeling with all the tweets. Still, there are too many, so instead of taking the original tweets of cop 21 and cop26, we only take the ones with retweets which are around 1/3 of the total but are the one needed to be propagated to the rest of the network.

Getting top influencer The top n influencers are simply the n influencers with the highest indegree in the retweet network, i.e., the most retweeted ones.

After doing this process for all the cops, the dataframes are merged and saved.

get topics At this point, we can run the topic modeling on the original tweets and then propagate the results to the retweets.

Chapter 5

Results

Let us start drawing a big picture of the different topics that emerged after the latent ideology analysis.

For cop26 topic modeling found 70 topics, the first step in the analysis was to remove all the topics with less than 2000 users, after the first filtering the topics left are 46.

After the latent ideology analysis some labels have been discarded because there were not enought edges to compute a statistically significative analysis. The final number of topics is 26.

At this point for cop26 we have assigned a latent ideology score to 1557 influencers and 22161 users on 26 topics.

The mean number of users with a ideology score for each topic is 1311, with a min of 151 and a max of 7764. Fig 5.1 contains a reap of all this results for both COP26 and COP21

Description	**COP21**	**COP26**
Initial topics	36	70
Topics >2000 users	18	46
Final topics	4	26
Influencers scored	270	1557
Users scored	7931	22161
Mean users/topic	2058	1311
Min users/topic	35	151
Max users/topic	7524	7764

Table 5.1 Summary of Latent ideology

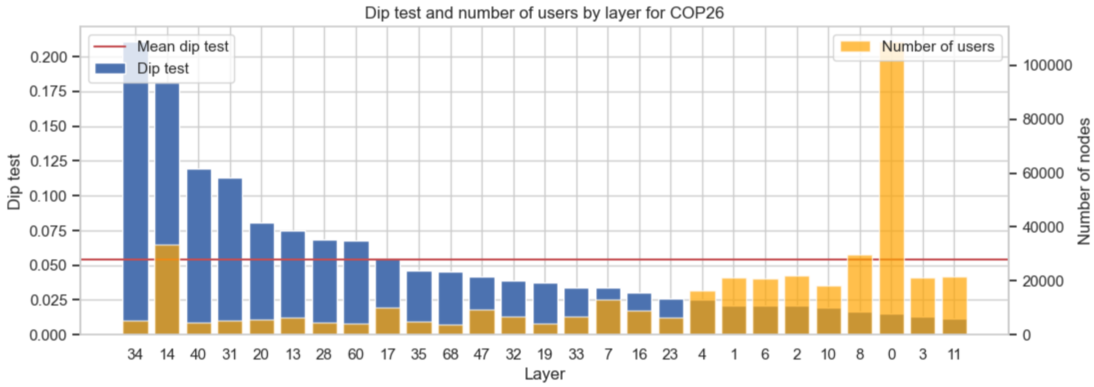

Fig. 5.1 Dip test and number of users for the layer with at least 2000 nodes

5.1 RQ1 Most Polarized Topics

Fig 5.1 summarize the size and the diptest result, so the polarization, for each topic. The first thing to note is the fact that to the highest polarization corresponds the smaller topics, with the exception of topic 14. For the first to topics the polarization is significantly higher than the others.

Fig 5.6 let us visualize the most and the least polarized topics, every network represent the retweet network of the 100 biggest influencers, in the leftmost plot we can find the full network while in the rightmost only the influencers are present. In the most polarized topics we can clearly see how the influencers are almost equally split between the two poles. The color of the node depends on its ideology score

It is interesting to note in Fig 5.7 the distribution of the tweets of each topics over time during the cop, the dotted line marks the start end end date of COP26. The most polarized had interest only in few days losing quickly the interest. The opposite happens in the least polarized topics where the discussion is distributed over a longest timespan.

5.2 RQ2 Longitudinal analysis

5.3 RQ3 User polarization among different topics

After computing the polarization score for all users we can now analyze whether the the users are polarized in the same way among all the topics they were active in.

The number of users involved in this analysis is 22161 active in 26 topics. most of them (16141) were only active in one topic, the maximum is 23 and the average is 1.53 topics per user.

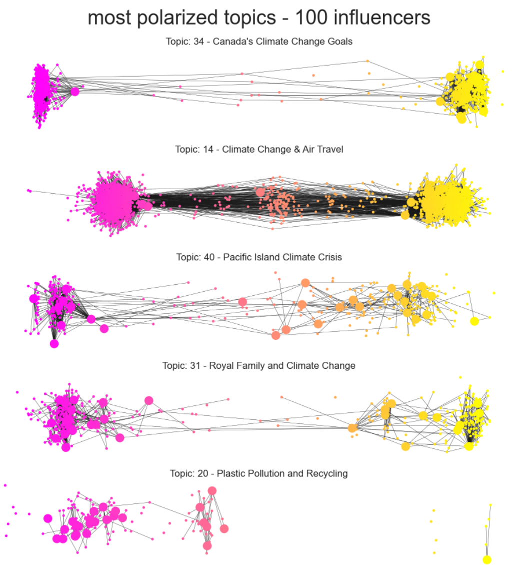

Fig. 5.2 most polarized topics

Fig. 5.3 most polarized topics only infleuncers

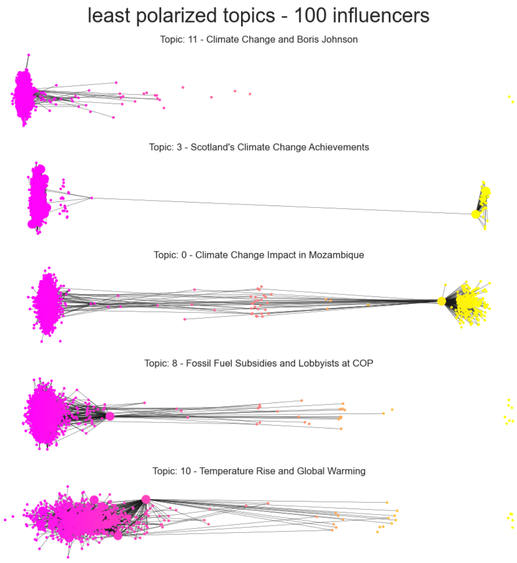

Fig. 5.4 least polarized topics

Fig. 5.5 least polarized topics only influencers

Fig. 5.6 least and most polarized topics

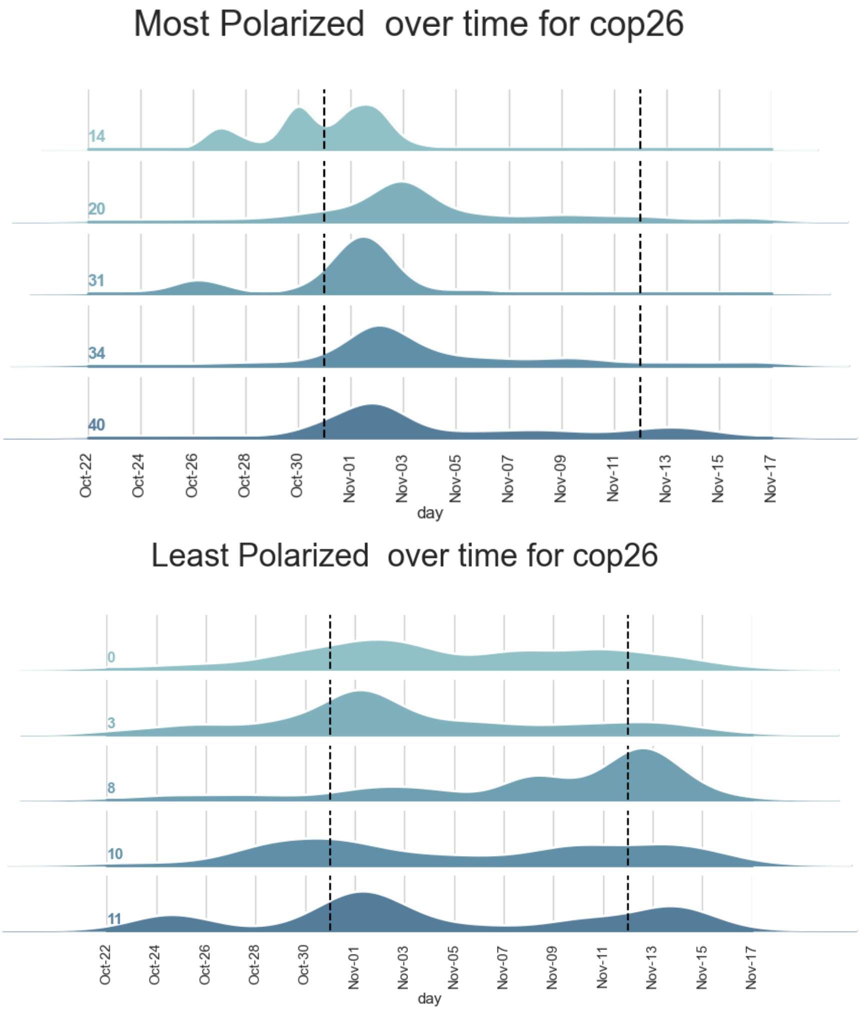

Fig. 5.7 Enter Caption

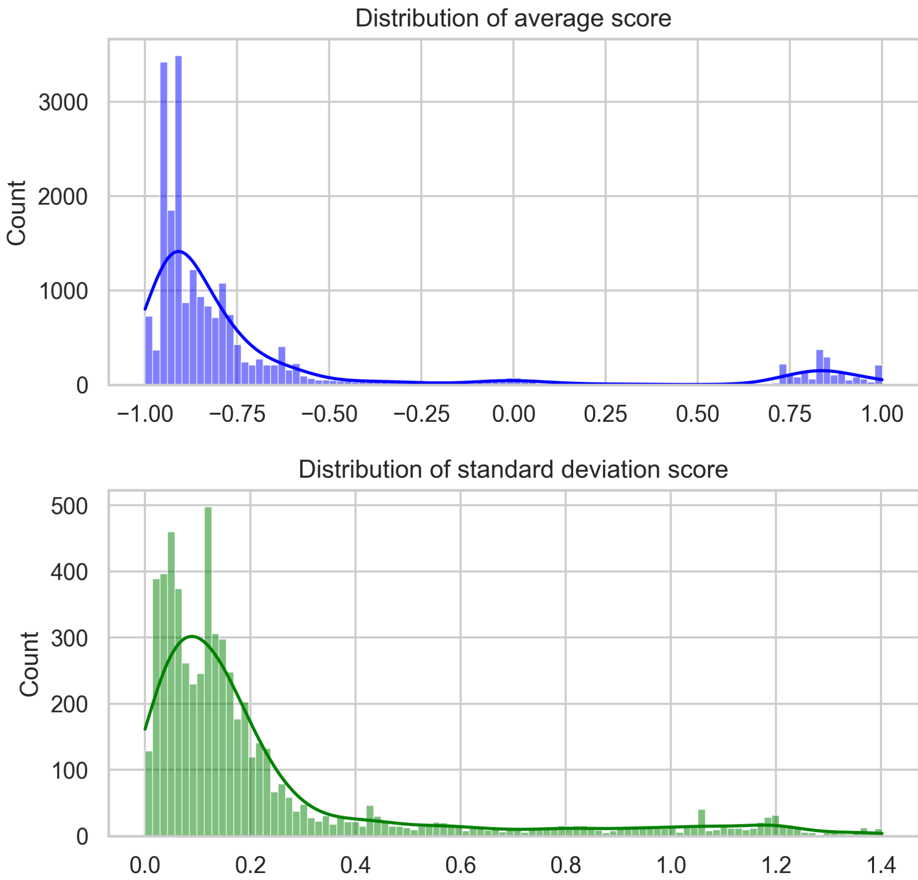

Fig. 5.8 Enter Caption

Then we computed, for each user present in more than 1 topic, the average and the standard deviation of the score. This value is higher for the users that are present in both side of the spectrum so this allow us to identify the degree to which users tend to be monopolar.

Fig 5.8 show how the distribution of the average score for every topic aggregated together, this matches with the global results of Falkenberg, where a majority is present on the -1 side versus a minority in the 1 side.

In the std we can see how there is a strong tendency to stay in the same side of the spectrum.

5.4 RQ4 Polarization of experts vs know-it-all

Chapter 6

Conclusions

References

[1] Syukuro Manabe and Richard T. Wetherald. Thermal equilibrium of the atmosphere with a given distribution of relative humidity. 24(3):241–259. ISSN 0022-4928, 1520-0469. doi: 10.1175/1520-0469(1967)024<0241:TEOTAW>2.0.CO;2. URL https://journals.ametsoc.org/view/journals/atsc/24/3/1520-0469_1967_024_0241_teotaw_2_0_co_2.xml.

[2] Charles D. Keeling, Robert B. Bacastow, Arnold E. Bainbridge, Carl A. Ekdahl, Peter R. Guenther, Lee S. Waterman, and John F. S. Chin. Atmospheric carbon dioxide variations at mauna loa observatory, hawaii. 28(6):538–551. ISSN 00402826, 21533490. doi: 10.1111/j.2153-3490.1976.tb00701.x. URL http://tellusa.net/index.php/tellusa/article/view/11322.

[3] F.W.G. Baker. First session of wmo–unep intergovernmental panel on climate change (ipcc), held in geneva, switzerland, during 9–11 november 1988. *Environmental Conservation*, 16(2):185–186, 1989. doi: 10.1017/S0376892900009127.

[4] Rafael M. Santos and Reza Bakhshoodeh. Climate change/global warming/climate emergency versus general climate research: comparative bibliometric trends of publications. 7(11):e08219. ISSN 2405-8440. doi: 10.1016/j.heliyon.2021.e08219. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8571708/.

[5] Anandita Sabherwal, Matthew T. Ballew, Sander Van Der Linden, Abel Gustafson, Matthew H. Goldberg, Edward W. Maibach, John E. Kotcher, Janet K. Swim, Seth A. Rosenthal, and Anthony Leiserowitz. The greta thunberg effect: Familiarity with greta thunberg predicts intentions to engage in climate activism in the united states. 51 (4):321–333. ISSN 0021-9029, 1559-1816. doi: 10.1111/jasp.12737. URL https://onlinelibrary.wiley.com/doi/10.1111/jasp.12737.

[6] Falkenberg. Climate polarization is increasing on twitter. 12(12):1094–1095. ISSN 1758-6798. doi: 10.1038/s41558-022-01533-z. URL https://www.nature.com/articles/s41558-022-01533-z.

[7] Pew Research Center. Politics on twitter: One-third of tweets from u.s. adults are political.

[8] Naveen Kumar Arora and Isha Mishra. COP26: more challenges than achievements. 4(4):585–588. ISSN 2523-8922. doi: 10.1007/s42398-021-00212-7. URL https://doi.org/10.1007/s42398-021-00212-7.

[9] Juan Layna and Leandro Altamirano. Promises that don't work? COP26 and the problems of climate change. 8(3):118–129. ISSN 2413-8053. doi: 10.17351/ests2022.1377. URL https://estsjournal.org/index.php/ests/article/view/1377.

[10] K. Suresh. Climate change challenge 26th conference of parties (COP26) climate summit is crucial but may be disappointing? 6(1):100–104. doi: 10.17352/gje.000051. URL https://www.peertechzpublications.org/articles/GJE-6-151.php.

[11] Max Falkenberg, Alessandro Galeazzi, Maddalena Torricelli, Niccolò Di Marco, Francesca Larosa, Madalina Sas, Amin Mekacher, Warren Pearce, Fabiana Zollo, Walter Quattrociocchi, and Andrea Baronchelli. Growing polarization around climate change on social media. *Nature Climate Change*, 12(12):1114–1121, December 2022. ISSN 1758-6798. doi: 10.1038/s41558-022-01527-x. URL https://www.nature.com/articles/s41558-022-01527-x.

[12] Aaron Bramson, Patrick Grim, Daniel J. Singer, William J. Berger, Graham Sack, Steven Fisher, Carissa Flocken, and Bennett Holman. Understanding polarization: Meanings, measures, and model evaluation. 84 (1):115–159. ISSN 0031-8248, 1539-767X. doi: 10.1086/688938. URL https://www.cambridge.org/core/journals/philosophy-of-science/article/abs/understanding-polarization-meanings-measures-and-model-evaluation/FFACA58EFC4A6E421107020290BF592C.

[13] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3(null):993–1022, March 2003. ISSN 1532-4435.

[14] Daniel Lee and H. Sebastian Seung. Algorithms for Non-negative Matrix Factorization. In *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://papers.nips.cc/paper/2000/hash/f9d1152547c0bde01830b7e8bd60024c-Abstract.html.

[15] Da Kuang, Jaegul Choo, and Haesun Park. Nonnegative Matrix Factorization for Interactive Topic Modeling and Document Clustering. In M. Emre Celebi, editor, *Partitional Clustering Algorithms*, pages 215–243. Springer International Publishing, Cham, 2015. ISBN 9783319092591. doi: 10.1007/978-3-319-09259-1_7. URL https://doi.org/10.1007/978-3-319-09259-1_7.

[16] Xiaohui Yan, Jiafeng Guo, Yanyan Lan, and Xueqi Cheng. A biterm topic model for short texts. In *Proceedings of the 22nd international conference on World Wide Web*, WWW '13, pages 1445–1456, New York, NY, USA, May 2013. Association for Computing Machinery. ISBN 978-1-4503-2035-1. doi: 10.1145/2488388.2488514. URL https://doi.org/10.1145/2488388.2488514.

[17] Jianhua Yin and Jianyong Wang. A dirichlet multinomial mixture model-based approach for short text clustering. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 233–242, August 2014. doi: 10.1145/2623330.2623715. URL https://dl.acm.org/doi/10.1145/2623330.2623715.

[18] Maarten Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure, March 2022. URL http://arxiv.org/abs/2203.05794. arXiv:2203.05794 [cs].

[19] Dimo Angelov. Top2Vec: Distributed Representations of Topics, August 2020. URL http://arxiv.org/abs/2008.09470. arXiv:2008.09470 [cs, stat].

[20] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. URL http://arxiv.org/abs/1706.03762.

[21] Roman Egger and Joanne Yu. A Topic Modeling Comparison Between LDA, NMF, Top2Vec, and BERTopic to Demystify Twitter Posts. *Frontiers in Sociology*, 7:886498, May 2022. ISSN 2297-7775. doi: 10.3389/fsoc.2022.886498. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9120935/.

[22] Michael Steinbach, Levent Ertöz, and Vipin Kumar. The challenges of clustering high dimensional data. In Luc T. Wille, editor, *New Directions in Statistical Physics: Econophysics, Bioinformatics, and Pattern Recognition*, pages 273–309. Springer. ISBN 9783662089682. doi: 10.1007/978-3-662-08968-2_16. URL https://doi.org/10.1007/978-3-662-08968-2_16.

[23] Andrzej Maćkiewicz and Waldemar Ratajczak. Principal components analysis (PCA). 19(3):303–342. ISSN 0098-3004. doi: 10.1016/0098-3004(93)90090-R. URL https://www.sciencedirect.com/science/article/pii/009830049390090R.

[24] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[25] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Großberger. Umap: Uniform manifold approximation and projection. *Journal of Open Source Software*, 3(29):861, 2018. doi: 10.21105/joss.00861. URL https://doi.org/10.21105/joss.00861.

[26] Xin Jin and Jiawei Han. K-means clustering. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 563–564. Springer US. ISBN 9780387301648. doi: 10.1007/978-0-387-30164-8_425. URL https://doi.org/10.1007/978-0-387-30164-8_425.

[27] Leland McInnes, John Healy, and Steve Astels. hdbscan: Hierarchical density based clustering. 2(11):205. ISSN 2475-9066. doi: 10.21105/joss.00205. URL https://joss.theoj.org/papers/10.21105/joss.00205.

[28] Data mining. In Anand Rajaraman and Jeffrey David Ullman, editors, *Mining of Massive Datasets*, pages 1–17. Cambridge University Press. ISBN 9781107737419. doi: 10.1017/CBO9781139058452.002. URL https://www.cambridge.org/core/books/mining-of-massive-datasets/data-mining/E5BFF4C1DD5A1FB946D616D619B373C2.

[29] Ishan Misra, C. Lawrence Zitnick, Margaret Mitchell, and Ross Girshick. Seeing through the human reporting bias: Visual classifiers from noisy human-centric labels. pages 2930–2939. URL https://www.cv-foundation.org/openaccess/content_cvpr_2016/html/Misra_Seeing_Through_the_CVPR_2016_paper.html.

[30] Luke Haliburton, Sinksar Ghebremedhin, Robin Welsch, Albrecht Schmidt, and Sven Mayer. Investigating labeler bias in face annotation for machine learning. URL http://arxiv.org/abs/2301.09902.

[31] Stephen P. Borgatti, Ajay Mehra, Daniel J. Brass, and Giuseppe Labianca. Network analysis in the social sciences. 323(5916):892–895. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.1165821. URL https://www.science.org/doi/10.1126/science.1165821.

[32] M. Kivela, A. Arenas, M. Barthelemy, J. P. Gleeson, Y. Moreno, and M. A. Porter. Multilayer networks. 2(3):203–271. ISSN 2051-1310, 2051-1329. doi: 10.1093/comnet/cnu016. URL https://academic.oup.com/comnet/article-lookup/doi/10.1093/comnet/cnu016.

[33] Jennifer McCoy, Tahmina Rahman, and Murat Somer. Polarization and the global crisis of democracy: Common patterns, dynamics, and pernicious consequences for democratic polities. *American Behavioral Scientist*, 62(1):16–42, 2018. doi: 10.1177/0002764218759576. URL https://doi.org/10.1177/0002764218759576.

[34] Michael Conover, Jacob Ratkiewicz, Matthew Francisco, Bruno Goncalves, Filippo Menczer, and Alessandro Flammini. Political polarization on twitter. 5(1):89–96. ISSN 2334-0770. doi: 10.1609/icwsm.v5i1.14126. URL https://ojs.aaai.org/index.php/ICWSM/article/view/14126.

[35] James Flamino, Alessandro Galezzi, Stuart Feldman, Michael W. Macy, Brendan Cross, Zhenkun Zhou, Matteo Serafino, Alexandre Bovet, Hernan A. Makse, and Boleslaw K. Szymanski. Shifting Polarization and Twitter News Influencers between two U.S. Presidential Elections. 2021. doi: 10.48550/ARXIV.2111.02505. URL https://arxiv.org/abs/2111.02505.

[36] Noam Gidron, James Adams, and Will Horne. Toward a comparative research agenda on affective polarization in mass publics. URL https://papers.ssrn.com/abstract=3391062.

[37] Hywel T. P. Williams, James R. McMurray, Tim Kurz, and F. Hugo Lambert. Network analysis reveals open forums and echo chambers in social media discussions of climate change. 32:126–138. ISSN 0959-3780. doi: 10.1016/j.gloenvcha.2015.03.006. URL https://www.sciencedirect.com/science/article/pii/S0959378015000369.

[38] Pablo Barberá. Birds of the Same Feather Tweet Together: Bayesian Ideal Point Estimation Using Twitter Data. *Political Analysis*, 23(1):76–91, January 2015. ISSN 1047-1987, 1476-4989. doi: 10.1093/pan/mpu011. URL https://www.cambridge.org/core/journals/political-analysis/article/birds-of-the-same-feather-tweet-together-bayesian-ideal-point-estimation-using-twitter-data/91E37205F69AEA32EF27F12563DC2A0A.

[39] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5):403–420, 1970. URL https://ieeexplore.ieee.org/document/1102314.

[40] J. A. Hartigan and P. M. Hartigan. The Dip Test of Unimodality. *The Annals of Statistics*, 13(1):70–84, March 1985. ISSN 0090-5364, 2168-8966. doi: 10.1214/aos/1176346577. URL https://projecteuclid.org/journals/annals-of-statistics/volume-13/issue-1/The-Dip-Test-of-Unimodality/10.1214/aos/1176346577.full.

[41] Silvia Terragni and Elisabetta Fersini. OCTIS 2.0: Optimizing and Comparing Topic Models in Italian Is Even Simpler! In Elisabetta Fersini, Marco Passarotti, and Viviana Patti, editors, *Proceedings of the Eighth Italian Conference on Computational Linguistics, CLiC-it 2021, Milan, Italy, January 26-28, 2022*, volume 3033 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2021. URL http://ceur-ws.org/Vol-3033/paper55.pdf.

[42] Silvia Terragni, Elisabetta Fersini, Bruno Giovanni Galuzzi, Pietro Tropeano, and Antonio Candelieri. OCTIS: Comparing and optimizing topic models is simple! In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 263–270. Association for Computational Linguistics, April 2021. URL https://www.aclweb.org/anthology/2021.eacl-demos.31.

[43] Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. ClimateBert: A Pretrained Language Model for Climate-Related Text, December 2022. URL http://arxiv.org/abs/2110.12010. arXiv:2110.12010 [cs].

[44] Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder. URL http://arxiv.org/abs/1803.11175.

[45] Alexander Hoyle, Pranav Goel, Denis Peskov, Andrew Hian-Cheong, Jordan Boyd-Graber, and Philip Resnik. Is Automated Topic Model Evaluation Broken?: The Incoherence of Coherence, October 2021. URL http://arxiv.org/abs/2107.02173. arXiv:2107.02173 [cs].

[46] Davide Vega and Matteo Magnani. Foundations of Temporal Text Networks. *Applied Network Science*, 3(1):1–26, December 2018. ISSN 2364-8228. doi: 10.1007/s41109-018-0082-3. URL https://appliednetsci.springeropen.com/articles/10.1007/s41109-018-0082-3.

Appendix A

How to install LaTeX

Windows OS

TeXLive package - full version

1. Download the TeXLive ISO (2.2GB) from https://www.tug.org/texlive/
2. Download WinCDEmu (if you don't have a virtual drive) from http://wincdemu.sysprogs.org/download/
3. To install Windows CD Emulator follow the instructions at http://wincdemu.sysprogs.org/tutorials/install/
4. Right click the iso and mount it using the WinCDEmu as shown in http://wincdemu.sysprogs.org/tutorials/mount/
5. Open your virtual drive and run setup.pl

or

Basic MikTeX - TeX distribution

1. Download Basic-MiKTeX(32bit or 64bit) from http://miktex.org/download
2. Run the installer
3. To add a new package go to Start » All Programs » MikTex » Maintenance (Admin) and choose Package Manager

4. Select or search for packages to install

TexStudio - TeX editor

1. Download TexStudio from
   http://texstudio.sourceforge.net/#downloads

2. Run the installer

Mac OS X

MacTeX - TeX distribution

1. Download the file from
   https://www.tug.org/mactex/

2. Extract and double click to run the installer. It does the entire configuration, sit back and relax.

TexStudio - TeX editor

1. Download TexStudio from
   http://texstudio.sourceforge.net/#downloads

2. Extract and Start

Unix/Linux

TeXLive - TeX distribution

Getting the distribution:

1. TexLive can be downloaded from
   http://www.tug.org/texlive/acquire-netinstall.html.

2. TexLive is provided by most operating system you can use (rpm,apt-get or yum) to get TexLive distributions

Installation

1. Mount the ISO file in the mnt directory

   ```
   mount -t iso9660 -o ro,loop,noauto /your/texlive####.iso /mnt
   ```

2. Install wget on your OS (use rpm, apt-get or yum install)

3. Run the installer script install-tl.

   ```
   cd /your/download/directory
   ./install-tl
   ```

4. Enter command ‘i’ for installation

5. Post-Installation configuration:
   http://www.tug.org/texlive/doc/texlive-en/texlive-en.html#x1-320003.4.1

6. Set the path for the directory of TexLive binaries in your .bashrc file

For 32bit OS

For Bourne-compatible shells such as bash, and using Intel x86 GNU/Linux and a default directory setup as an example, the file to edit might be

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/i386-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

For 64bit OS

```
edit $~/.bashrc file and add following lines
PATH=/usr/local/texlive/2011/bin/x86_64-linux:$PATH;
export PATH
MANPATH=/usr/local/texlive/2011/texmf/doc/man:$MANPATH;
export MANPATH
```

```
INFOPATH=/usr/local/texlive/2011/texmf/doc/info:$INFOPATH;
export INFOPATH
```

Fedora/RedHat/CentOS:

```
sudo yum install texlive
sudo yum install psutils
```

SUSE:

```
sudo zypper install texlive
```

Debian/Ubuntu:

```
sudo apt-get install texlive texlive-latex-extra
sudo apt-get install psutils
```

Appendix B

Installing the CUED class file

LaTeX.cls files can be accessed system-wide when they are placed in the <texmf>/tex/latex directory, where <texmf> is the root directory of the user's TeXinstallation. On systems that have a local texmf tree (<texmflocal>), which may be named "texmf-local" or "localtexmf", it may be advisable to install packages in <texmflocal>, rather than <texmf> as the contents of the former, unlike that of the latter, are preserved after the LaTeXsystem is reinstalled and/or upgraded.

It is recommended that the user create a subdirectory <texmf>/tex/latex/CUED for all CUED related LaTeXclass and package files. On some LaTeXsystems, the directory look-up tables will need to be refreshed after making additions or deletions to the system files. For TeXLive systems this is accomplished via executing "texhash" as root. MIKTeXusers can run "initexmf -u" to accomplish the same thing.

Users not willing or able to install the files system-wide can install them in their personal directories, but will then have to provide the path (full or relative) in addition to the filename when referring to them in LaTeX.