



UPPSALA  
UNIVERSITET

IT 22 133

Thesis 30 Credits

October 2022

# Tracking the Conversation around NATO in the Nordic States Using Machine Learning

---

Benedict Treeby

Master's Programme in Data Science  
(Machine Learning & Statistics)







UPPSALA  
UNIVERSITET

## Tracking the Conversation around NATO in the Nordic States Using Machine Learning

---

Benedict Treeby

### **Abstract**

Online communication is makes up a large part of the public discourse around current events, and increasingly has been the target for disinformation campaigns. Clearly, there is a need for some language-agnostic tool to map how the discourse shifts and evolves. As such, the goal of this analysis was to create a tool to collate posts from Facebook, Instagram, Reddit and Twitter into one dataset for comparison and analysis, then to generate weekly topics based off posts containing the term "NATO" from this dataset. In this way BERTopic was used in an attempt to divide posts relating to "NATO" into these topics and then map them week-by-week, creating connections between weeks based on the overall topic. This period ranged from the buildup of the Ukrainian/Russian conflict (1 November) until 2 June. The analysis covers four Nordic languages; Danish, Finnish, Norwegian and Swedish from the aforementioned platforms. The Swedish model proved unfruitful due to the stochastic nature of the algorithm, however for the Danish, Finnish and Norwegian datasets, broad overall topics were generated as well as individual weekly topics for each of the broader topics.

**Faculty of Science and Technology**

**Uppsala University, Uppsala**

Supervisor: Davide Vega D'Aurelio Subject reader: Matteo Magnani

Examiner: Davide Vega D'Aurelio



## Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Background</b>	<b>5</b>
2.1	BERTopic . . . . .	5
2.1.1	BERT . . . . .	6
2.1.2	UMAP . . . . .	8
2.1.3	HDBSCAN . . . . .	11
2.1.4	c-TF-IDF . . . . .	15
2.1.5	BERTopic . . . . .	16
2.1.6	Advantages . . . . .	17
2.1.7	Disadvantages . . . . .	17
<b>3</b>	<b>Data</b>	<b>18</b>
3.1	Choice of Platforms . . . . .	18
3.2	Data Collection . . . . .	19
3.2.1	Twitter Structure . . . . .	19
3.2.2	CrowdTangle (Facebook & Instagram) Structure . . . . .	20
3.2.3	Reddit Structure . . . . .	22
3.3	Cleaning the Dataset . . . . .	23
3.3.1	Handling Shortened Links . . . . .	23
3.3.2	Ambiguous Links . . . . .	23
3.4	Duplicate entries . . . . .	24
3.5	The Dataset . . . . .	24
3.5.1	Example Data Subset . . . . .	26
3.5.2	Deleted posts . . . . .	27
3.5.3	Noise . . . . .	27
<b>4</b>	<b>Methodology</b>	<b>29</b>
4.1	Data Collection Tools . . . . .	29
4.1.1	Twitter API . . . . .	30
4.1.2	Facebook and Instagram API . . . . .	30
4.1.3	Reddit API . . . . .	30
4.2	Searching . . . . .	31
4.2.1	Keyword Search . . . . .	32
4.2.2	Boolean Search . . . . .	32

4.3	Duplicate Entries . . . . .	33
4.4	Preprocessing . . . . .	33
4.5	Analysis . . . . .	33
4.5.1	BERTopic . . . . .	33
4.5.2	Stopword Removal . . . . .	34
4.5.3	Topic Reduction . . . . .	34
4.5.4	Topics Over Time . . . . .	34
<b>5</b>	<b>Results</b>	<b>35</b>
5.1	Raw Topics Before Topic Reduction . . . . .	35
5.2	Topic Reduction . . . . .	39
5.3	Further Topic Reduction . . . . .	41
5.4	Topics Over Time . . . . .	44
<b>6</b>	<b>Conclusions</b>	<b>49</b>
6.1	Limitations and Caveats . . . . .	50
6.1.1	Facebook/Instagram . . . . .	50
6.1.2	Reddit . . . . .	50
6.1.3	Twitter . . . . .	51
6.1.4	Data Collection . . . . .	51
6.1.5	BERT . . . . .	51
6.1.6	BERTopic . . . . .	51
6.2	Future Work . . . . .	52
6.2.1	Data Collection . . . . .	52
6.2.2	Dataset . . . . .	52
6.2.3	NLP techniques . . . . .	53
6.2.4	Optimizing the Algorithm . . . . .	53
6.2.5	Semi-Supervised Topic Modelling . . . . .	53
6.2.6	Further Investigation Into Results . . . . .	54
6.3	Ethical Implications . . . . .	54
<b>7</b>	<b>Appendix</b>	<b>58</b>

## 1 Introduction

Online social media platforms are becoming increasingly popular and therefore, increasingly the location where discourse surrounding current events occurs. As both online misinformation (misleading information) and disinformation (false information) are increasingly areas of concern, there exists a need to map the general sentiment of these conversations and their evolution. By doing this it paves the way for a qualitative analysis of how events impact the online discussion. However this mapping is obviously a non-trivial task, made more complex by the plethora of languages and geographical locations that divide these conversations. This means that a solution must not only be able to track specific topics over time, but also must be language agnostic.

Tracking the evolution of conversations allows us to be able to produce weekly topics to qualitatively assess the extent to which popular opinion are being swayed by so called “fake news”, public smear campaigns, or one-sided. This in turn allows for better awareness and provides valuable insight into how to counteract these attempts to sway public opinion.

The problem; is it possible to track the online conversation around a specific word or phrase, in this case “NATO” on social media in Danish, Finnish, Norwegian and Swedish either quantitatively or qualitatively. To simplify this problem, it is enough to say that tracking the shifting conversations over time is equivalent to finding topics over time that model the conversation. While it is likely that there is some crossover between the discourse across the different Nordic languages, that is beyond the scope of the exercise.

There have been previous attempts to create such a model, for example, Blei and Lafferty’s 2006 paper [1] presents a Gaussian model as a potential solution to model topics over time. However, more recent advances has come up with more modern solutions like that used here, the algorithm developed by Grootendorst, BERT (Bidirectional Encoder Representations from Transformers) in his 2022 paper [2].

## 2 Background

### 2.1 BERTopic

BERTopic was developed by Maarten Grootendorst [2]. The method uses BERT to create sentence-level embeddings for each document. The basic idea is the following: use BERT [3] to create context-aware, sentence-level embeddings for each document, then take the average of these embeddings so that each document is represented by a vector. Then using these vectors, apply the UMAP algorithm to project the data onto a lower dimension. Once this has been achieved then the data may be clustered using HDBSCAN and the c-TF-IDF scores are calculated, the clusters now corresponding to a class.

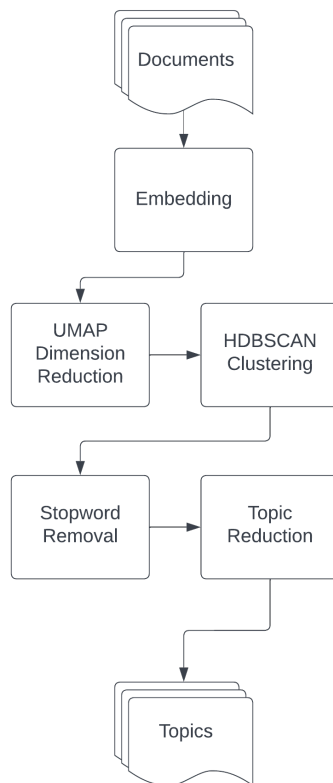


Figure 1: BERTopic overview

A brief overview of the methods used in the BERTopic method is provided herein; it is not intended to be exhaustive, but rather, a broad, holistic overview. Each step of the algorithm is displayed in Figure 1. Each box is a step of the algorithm and is explained further within the text.

### 2.1.1 BERT

Bidirectional Encoder Representations from Transformers (BERT) [3] is a semi-supervised natural language processing model specifically designed to create context-aware embeddings of language. The original paper provided two separate models; **BERT<sub>BASE</sub>** and **BERT<sub>LARGE</sub>**. **BERT<sub>BASE</sub>** contains 12 layers, 768 nodes per layer, 12 self-attention heads and 110M total parameters, this model is shown in Figure 2. The original BERT model was trained on a large corpus of text with the following two separate tasks.

- **Task 1: Masked Language Modelling** The goal here is given some input text and having masked some random words within that text, predict the masked words based of the surrounding words in the text. This process is to ensure that the encodings created are aware of context (This is also referred to as a Cloze task). For example, imagine that we have the sentence; “I took the dog for a walk”, this sentence may be transformed into “I took the \_ for a walk”, with the word “dog” being masked. The model is then trained by predicting the masked word, updating the model based on the outcome.
- **Task 2: Next Sentence Prediction** Given two sentences *A* and *B*, predict if *B* follows *A*.

Unlike another model like Word2Vec [4], BERT uses the input while both training/generating the model and in the resultant output model. A quite popular example to illustrate this is the following:

**A** - They robbed the bank

**B** - They sat down by the river bank

The word “bank” in **A** has a different meaning to the “bank” in **B**. As BERT creates contextualised embeddings the outputs for each sentence will be different. In another model like Word2Vec which is merely built using

context, these embeddings would be very similar. It is very easy to see from this example that context is incredibly important to understanding language, without the ability detect context the model would have difficulty understanding homonyms.

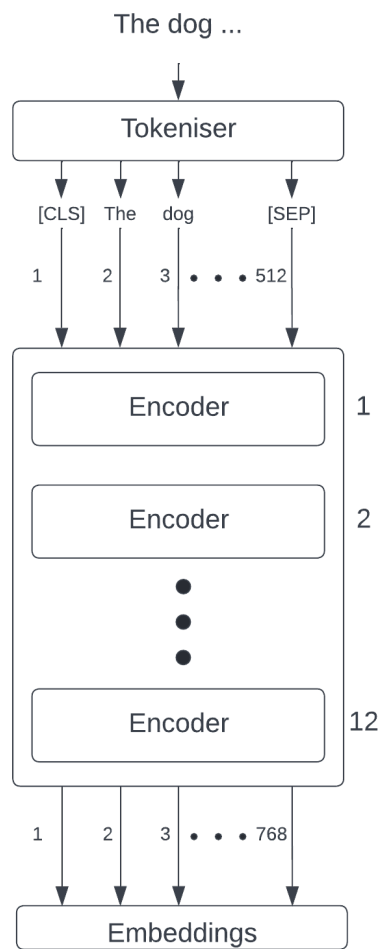


Figure 2: BERT model structure

### 2.1.2 UMAP

Uniform Manifold Approximation and Projection (UMAP) [5] is a dimension reduction algorithm designed to reduce the dimensionality of the input data while maintaining the relationships that the data contains in the original space. For example, UMAP is very good at preserving the global structure of large, high dimensional datasets.

UMAP is very similar in implementation to t-distributed stochastic neighbour embedding (t-SNE) [6] and similarly has non-deterministic behaviour. However, its runtime scales far better than t-SNE for larger datasets, as seen in Figure 3. The basic idea behind the algorithm is to construct a graph of the data, with each node representing a single point of data. Then try to recreate that graph in lower dimensions while retaining the same structure. To build this higher dimensional graph we create a weighted graph with the edge weights corresponding to the likelihood that any two particular nodes are connected. A point is “connected” if it’s within a locally calculated radius depending on the distance to the  $n^{\text{th}}$  nearest neighbour. By calculating the radius locally for each point UMAP can handle clusters of different densities. The “fuzziness” comes from UMAP decreasing the likelihood of a connection as the radius grows. To maintain the local structure of the data the algorithm dictates that each point is at least connected to its closest neighbour.

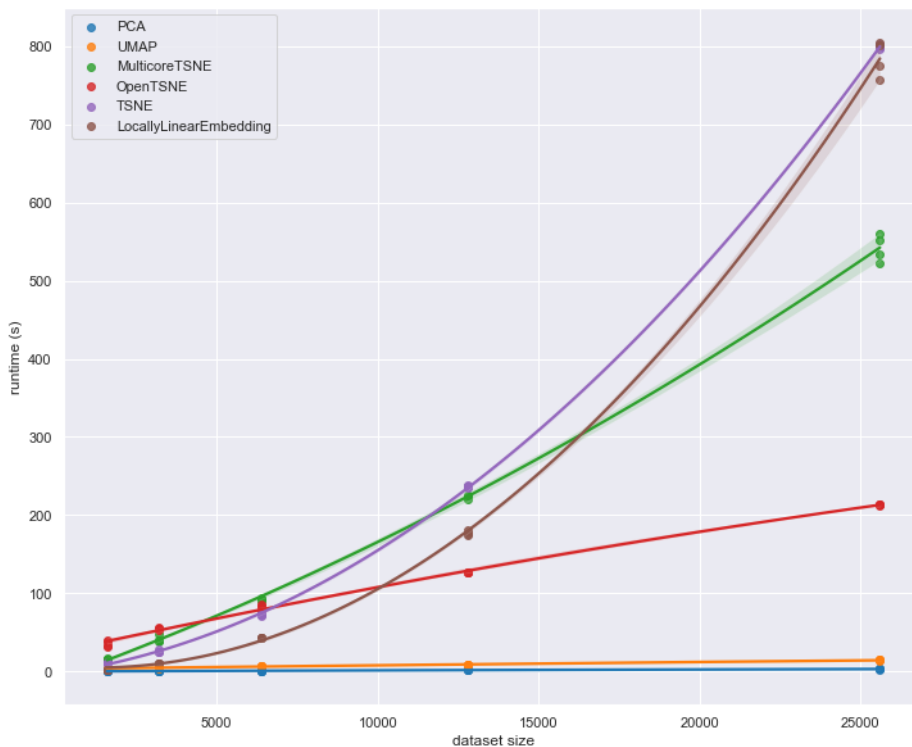


Figure 3: Performance comparisons of UMAP and other competing algorithms using the MNIST dataset [7][8]

UMAP has a few interesting idiosyncrasies that need to be addressed. The first is that distance within the projection may not mean anything at all. For example, take the projection of a mammoth in 3D to 2D in Fig. 4. The lower half of one of the hind legs is split into two separate and distinct clusters in the projection. Another quirk is that since UMAP is stochastic it may be that a particular projection is poor and the algorithm needs to be run several times to generate a reasonable result. Of course determining if this projection is “reasonable” can be difficult with higher dimensional data.

It has two major parameters. The first, **m\_distance**, this number controls the topology within local structures. If this number is too low as in Fig. 5a then too much local structure is lost, the space between each datapoint in

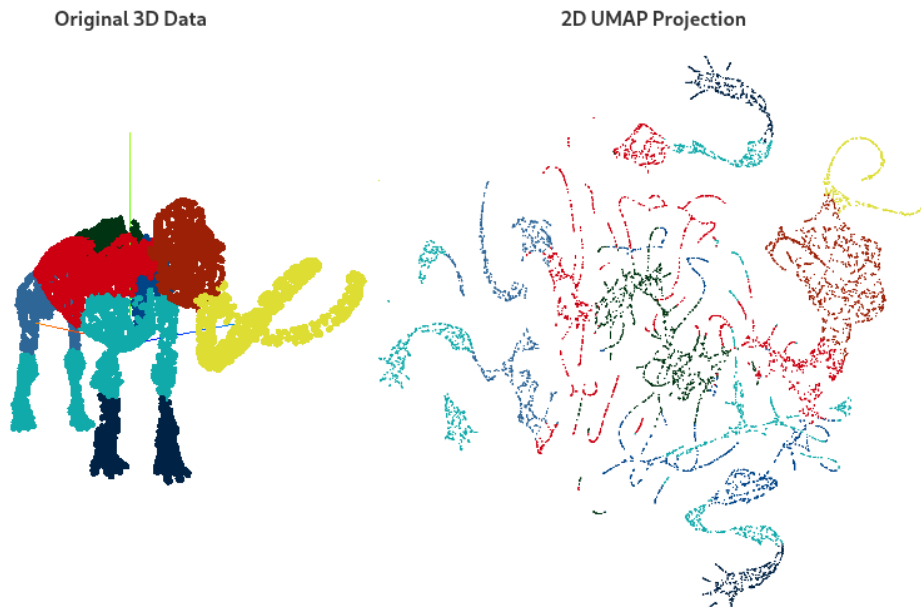


Figure 4: UMAP projection of 3D data onto 2D space [6]

the various localities becoming too small as compared to that within Fig. 4. On the other hand, if this parameter is too large as in Fig. 5b then the local structure is weighted too highly, ignoring the global structure of the data. The second parameter, **n\_neighbours**, manages the projection's global structure. If this parameter is too high as in Fig. 5c then too much local structure is lost. Each locality is clearly defined in relation to the others but the space within the localities and the shape of each locality has lost information when compared to the projection in Fig. 4. When both these parameters are too high as in Fig. 5d then in this specific case a projection is made that as a human, with highlighted clusters the individual clusters and shape of the original of the 3D structure is easily apparent. However, if these localities were not already highlighted then it would be exceedingly difficult to discover many of these clusters either manually or by some algorithm. The key here is to find some value where the detail is preserved while also creating enough structure to provide a coherent result.

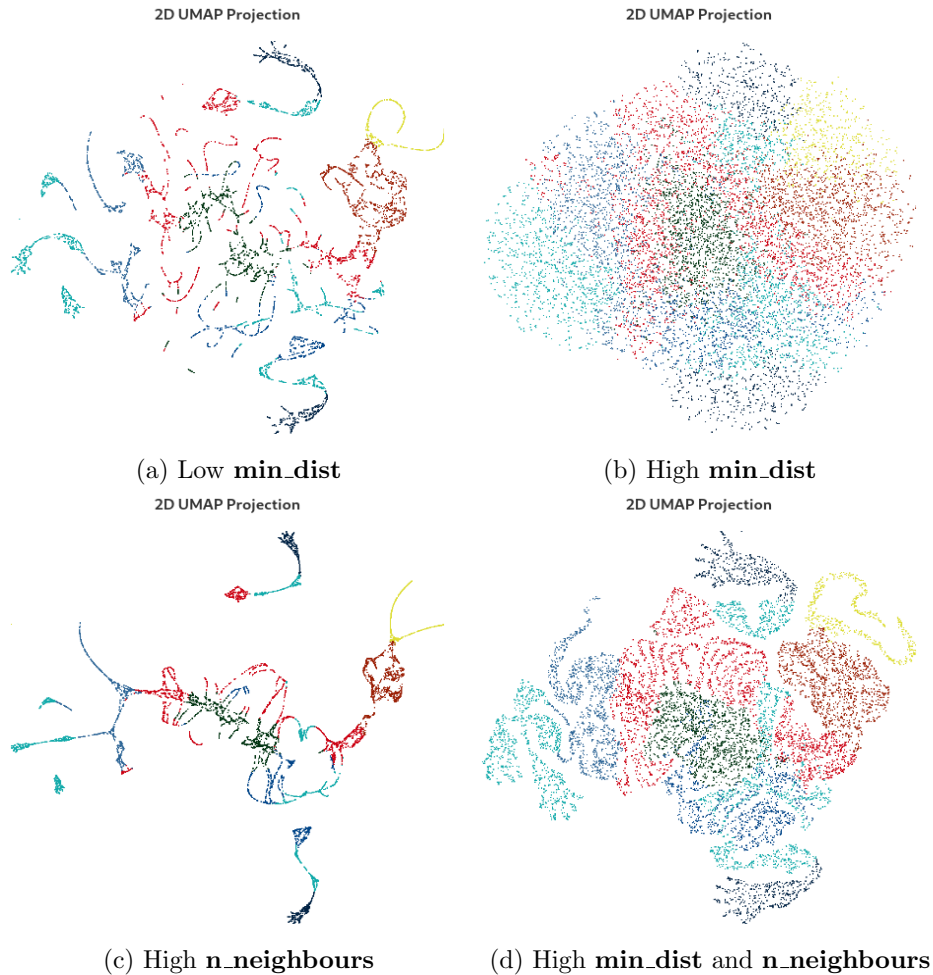


Figure 5: UMAP projections of a 3D mammoth with various changes in the **min\_dist** and **n\_neighbours** parameters

### 2.1.3 HDBSCAN

Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) is a clustering algorithm developed by McInnes et al. [9]. It was designed to accommodate noisy data and clusters of varying shapes and densities. The algorithm aims to find points of relative density and location and cluster them together. The code used to generate the images was also

written by McInnes et al. [10].

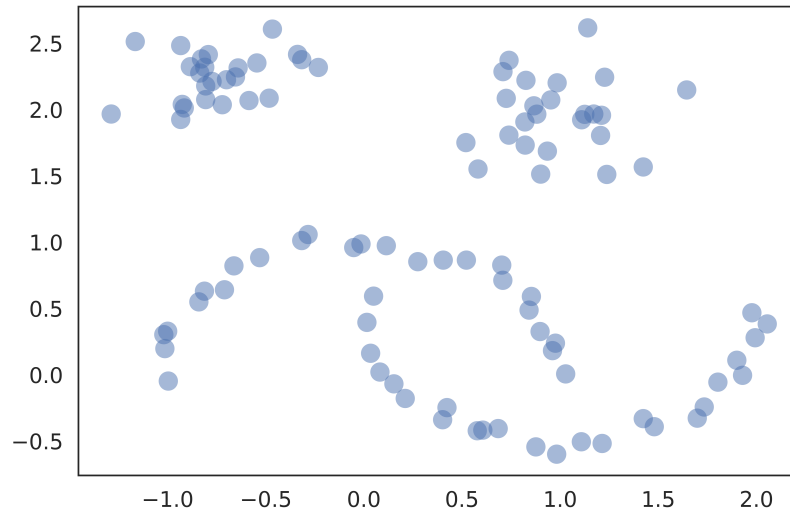


Figure 6: Sample data to cluster

Before considering the algorithm in full the mutual reachability distance needs to be defined (MRD).

Mutual reachability distance:

$$d_{mreach-k}(a, b) = \max\{\text{core}_k(a), \text{core}_k(b), d(a, b)\}$$

where  $\text{core}_k(a)$  is the distance from point  $a$  to its nearest  $k^{\text{th}}$  neighbour and  $d(a, b)$  is some defined distance metric between  $a$  and  $b$ , e.g Euclidean or cosine distance.

After calculating the MRD for each point (for example, the sample data in Figure 6) a Minimum Spanning Tree (MST) is built, with the edge weights between the nodes (corresponding to the datapoints) being equal to the MRD. Then edges with weight above some threshold are dropped. Doing

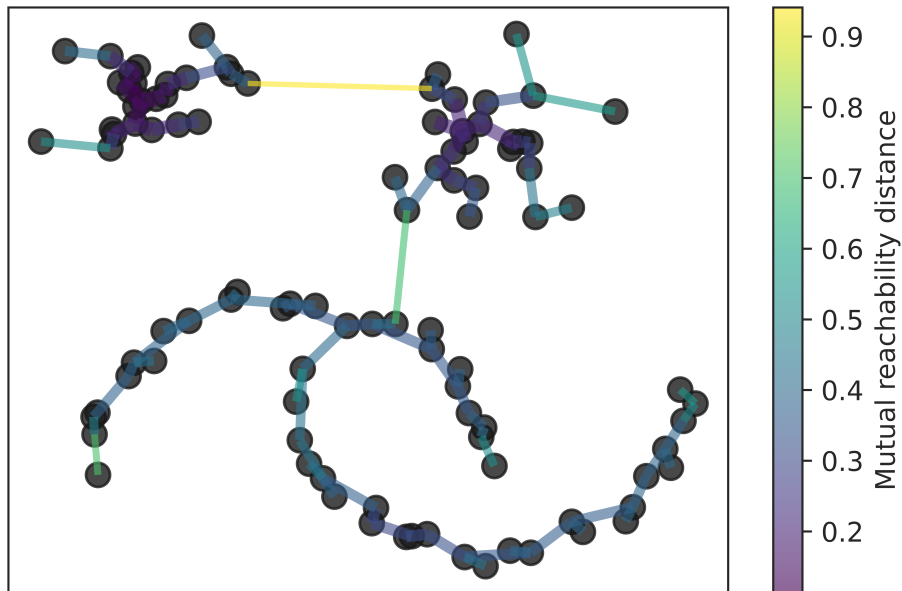


Figure 7: Minimum spanning tree

this process will create a graph of various isolated groups (components). This process is repeated again, dropping more edges above another, lower, threshold, until all there is no lower weight that could connect the separate components. This MST can be built using Prim's algorithm.

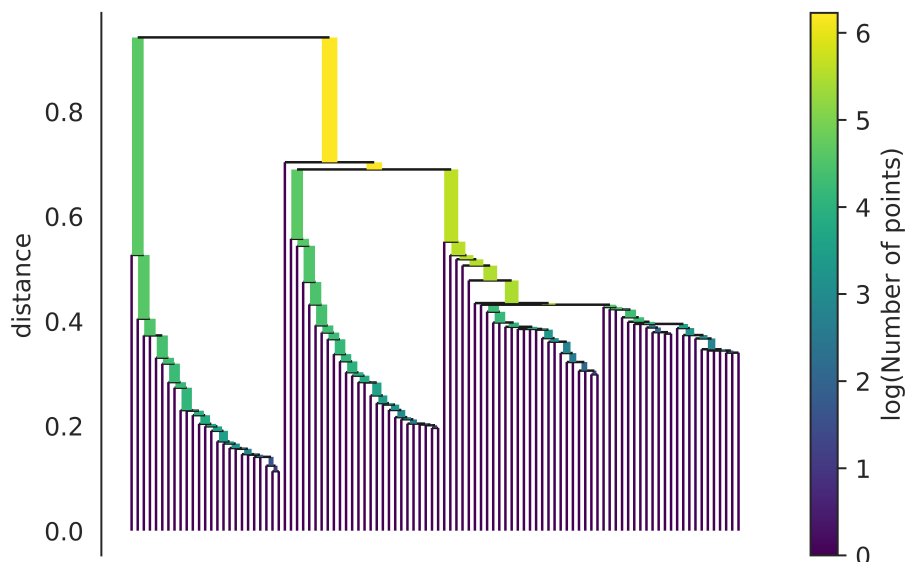


Figure 8: Dendrogram representing cluster hierarchy

Having created the MST (see Figure 7), a hierarchy of components is constructed (Figure 8). This is achieved by sorting the edges of the tree in ascending order, then for each edge create a new merged cluster. With that, there exists a clustered tree and the decision needs to be made where to cut the dendrogram. In a similar method like DBSCAN, it would merely cut through all the clusters at the same point, ignoring that some clusters may be more sparsely populated than others. HDBSCAN takes another approach; consider the two clusters created by each split in the cluster hierarchy dendrogram. If each cluster is above the parameter **min\_cluster\_size** then those are both “true” clusters and the remain in the dendrogram. If however, one or both of those clusters are smaller than the **min\_cluster\_size** then the smaller of those clusters is said to have fallen outside of a cluster and those points are discarded, with the larger of the two clusters formed by the split being subsumed by the parent. In this way, continuing through the whole dendrogram the clusters are condensed and a result like Fig. 9 is formed, with the width of each cluster being proportional to the number of points in it and the height proportional to the combined distance.

Now to select the clusters that are used, we merely select the clusters with the largest inked areas, subject to the constraint that if a child cluster is chosen then its parent cannot be selected.

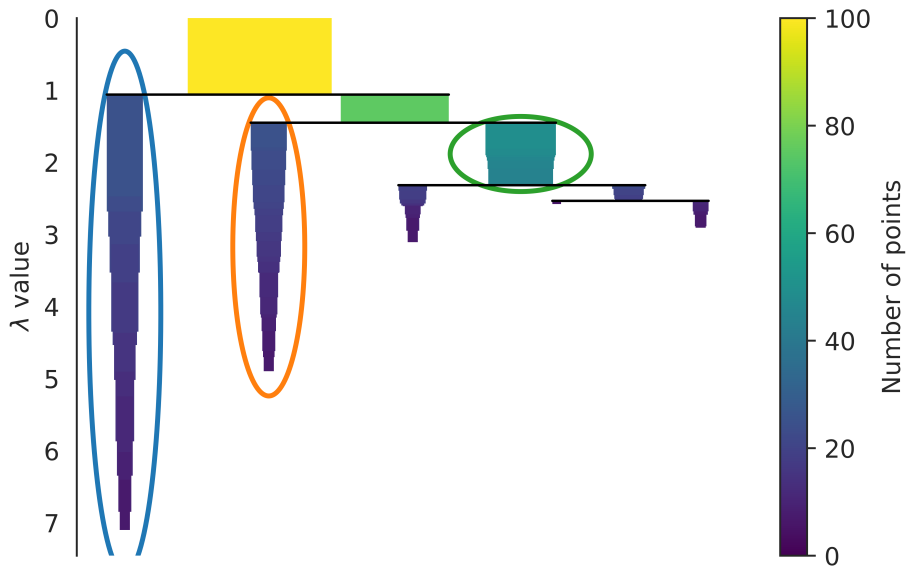


Figure 9: Condensed clusters

#### 2.1.4 c-TF-IDF

Class-based term frequency–inverse document frequency (c-TF-IDF) is the class-based score developed by Maarten Grootendorst for the BERTopic algorithm [2]. The goal is to generalize the concept of the TF-IDF score to groups of documents. Here for a class  $c$ , the term is represented by  $t$ , the frequency of the term in a class is  $f_{t,c}$ , that term’s frequency across all classes is represented by  $f_t$  and  $A$  is the average number of words per class.

$$W_{t,c} = f_{t,c} \cdot \log\left(1 + \frac{A}{f_t}\right)$$

There is also the temporal way of calculating these values, by introducing another variable  $i$  describing some time interval, which leverages the precal-

culated global IDF values as such;

$$W_{t,c,i} = tf_{t,c,i} \cdot \log\left(1 + \frac{A}{tf_t}\right)$$

This means the c-TF-IDF is calculated separately for each time interval  $i$ .

### 2.1.5 BERTopic

The topics generated by BERTopic will likely be full of so called “stop-words” e.g “and”, “or”, etc. These words add very little of meaning to the analysis so may be removed at this step. This is done by using a language-specific precompiled list of stopwords. This leads to topics made up of words not contained in the stopword list. This approach is not perfect however; misspelling or dialect-related differences can’t be accounted be accurately measured without great difficulty.

An optional step that could be applied involves reducing the number of topics. The method here is simple: merge the least frequent topic with the most similar topic as long as the corresponding c-TF-IDF scores cosine similarity are above 0.915 [11]. This repeats until the topics are too dissimilar to combine or the desired number of topics are achieved. Then the c-TF-IDF scores are recalculated. Now, finally the original documents are mapped into clusters, thereby forming topics to be used for the analysis.

An additional optional step which was applied in the analysis is described herein enables the mapping of the temporal development of topics. Bins are created corresponding to a time intervals of equal length. Each classes’ c-TD-IDF score is recalculated and the post’s timestamp allows consignment to the corresponding bin and topic. This change to the c-TF-IDF over time allows inferences to be drawn regarding a topic’s development over time. It is recommended to use less than 100 bins for both computational reasons and it is unlikely that there would be any significant development of a topic observable if the time interval was too short [11]. To say nothing of the amount of data that would be necessary to even create meaningful topics at such a fine level of granularity.

### 2.1.6 Advantages

**Number of topics** Traditional techniques such as Latent-Dirichlet Allocation (LDA) (proposed by Pritchard et al. [12] and expanded upon by Blei et al. [13]) have some shortcomings. Generally they require a pre-supposed number of topics, that is, we must assume that there is some fixed number of topics and then the model will find them. This can lead to the situation where there are either meaningless filler topics or meaningful topics are missed.

With BERTopic the number of topics is not fixed, the model merely clusters the data based off of the parameters provided to UMAP and HDBSCAN and produces some large number (potentially hundreds) of topics that are then able to be reduced (if desired).

**Context and Sentence Structure** A positive of of BERTopic When compared with similar topic modelling techniques like LDA or Blei and Lafferty’s 2006 Gaussian model is that context and sentence structure are preserved when constructing the topics. BERT’s context awareness capability means that, for example, the sentence “They fished down by the river bank” would be unlikely to produce the same output vector as “They robbed the bank” in BERT, thanks to its awareness of context. In both the LDA and GSDMM models this would not be the case since both sentences contained the word “bank”. Another advantage here is that BERTopic does not require that stopwords are removed before creating the embeddings; preprocessing actually worsens the performance of the method at this stage [14].

### 2.1.7 Disadvantages

**Clustering** One potential disadvantage of the method, is that due to HDBScan it is possible that some number of datapoints/documents remain unclustered. These points could be outliers, but they could also be part of some latent cluster in the data that the model hasn’t picked up on or be part of some natural formation that the BERT model is unable to parse correctly.

Whether or not this is actually a disadvantage or is in fact an advantage (due to the algorithm not creating spurious clusters) is open to debate, however it needs to be acknowledged that unclustered datapoints are a grey area in the analysis.

**UMAP** Despite the speed of the UMAP algorithm, the large dataset and complexity of the algorithm means it is unfeasible to perform parameter tuning. It is known that these parameters can make quite a difference and so by not optimising these values it possible that information is lost or not discovered when the dataset is reduced.

This issue of information loss may compounded with HDBScan, causing many more unclustered datapoints than there would be otherwise be the case.

**Hyperparameters** The choice of hyperparameters can have a large impact, as discussed. However, the choice of these parameters is difficult to make. The affect of these parameters on the model can be very difficult to discern with no “ground truth” to aim towards. That is, without some baseline it is difficult to determine if the change in parameters generates a model that more closely models reality.

## 3 Data

### 3.1 Choice of Platforms

Obviously, the best choices for platforms to explore are those that are the most popular. However, things are not that simple, as we are at the mercy of the platform. If the platform does not provide an access point for the data, then obtaining it manually becomes more difficult. The platforms that are readily accessible are Facebook, Instagram, and Twitter, which make up for over 90% of Sweden’s social media usage [15]. Pinterest was not chosen because it’s not a platform generally used to discuss news or politics. Reddit was included here because it has a relatively large Nordic presence. As of April 2022, r/Denmark had 301k subscribers, r/Finland

had 86k subscribers, r/Norway has 138k subscribers and r/Sweden has 386k subscribers respectively.

## 3.2 Data Collection

The dataset was built by searching for keywords provided by a third party to the research group under which this thesis is being done, the Uppsala University Information Laboratory. The specific keywords, phrases and URLs of interest were chosen with the intention of matching to posts that may contain misinformation surrounding the conflict in Ukraine. These keyword lists may be found in the appendices.

### 3.2.1 Twitter Structure

The fields below are saved by first calling a search function, in this instance, through the twitter API; where:

- **query** - User input field containing the query being made.
- **next\_token** - Response output when at least one additional page is found and the value is needed to fetch the following page.
- **start\_time** - User defined start date for the search.
- **end\_time** - User defined end date for the search. Omitting this variable sets this field to the default - the current time as of running the script.
- **tweet\_fields** - Fields to extract from each post; provided in a list in the format below.
- **max\_results** - User defined maximum number of posts to be returned.

```
tweet_fields=['created_at ', 'id ',  
             'author_id ',  
             'conversation_id ',  
             'in_reply_to_user_id ',  
             'referenced_tweets ',  
             'entities ',
```

```
        'public_metrics' ,  
        'source' ]
```

```
response = self.client.search_all_tweets(query=self.query ,  
                                         since_id=since_id ,  
                                         start_time=start_time ,  
                                         end_time=end_time ,  
                                         next_token=next_token ,  
                                         tweet_fields=tweet_fields ,  
                                         media_fields=['url'] ,  
                                         max_results=500)
```

### 3.2.2 CrowdTangle (Facebook & Instagram) Structure

For CrowdTangle, the request is conducted in steps. The user-input fields include:

- **self.search\_endpoint** - URL which is being queried, in this case, it is always: “https://api.crowdtangle.com/posts/search”.
- **pagination\_token** - Response output when at least one additional page is found and the value is needed to fetch the following page.
- **params** - Parameters contained in a dictionary and given directly to the API call. The contents are analysed below.
  - **token** - Personal API key of the account that is being using.
  - **count** - Number of posts to return, the maximum being 100.
  - **startDate** - The earliest date that a post is posted (in UTC).
  - **endDate** - The latest date that a post is posted (in UTC).

```
if pagination_token is None:  
    res = requests.get(self.search_endpoint , params)  
else:  
    res = requests.get(pagination_token)  
    if 200 > res.json()[ 'status ' ] > 300:  
        raise Exception()
```

```
res = json.loads(res.content.decode('utf-8'))['result']

params = {'token': self.api_key, 'count': '100',
'startDate': start_time,
'endDate': end_time}
```

The code here is the main loop to pull data from CrowdTangle. Each iteration checks if the response code for the request is healthy then sets the parameters for the next search.

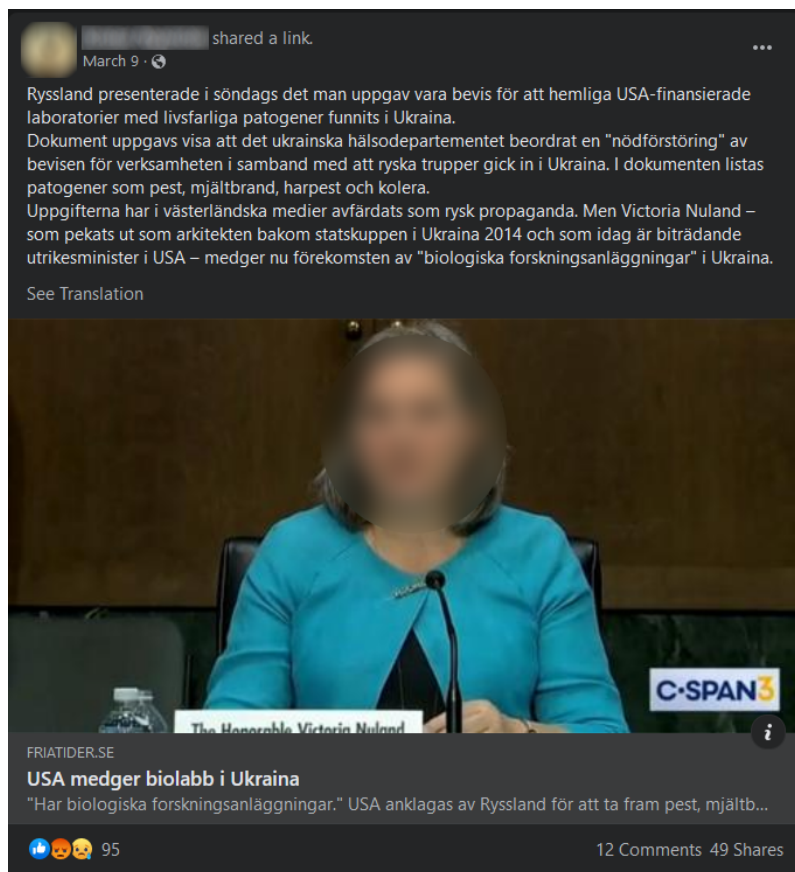


Figure 10: Example of a post on the platform Facebook.

Figure 10 shows an example Facebook post. It includes a text, a link that can be obtained, and various reactions in addition to the standard meta

data one would expect, i.e date posted, number of comments.

### 3.2.3 Reddit Structure

In Reddit's case, the query is a simple GET request that includes:

- **query** - contains the search query and similar to the website's search function.
- **sort** - Sorts posts according to this field's setting; "new" (newest to oldest) was the only option used throughout this exercise.
- **limit** - Number of posts to return; the API used doesn't use the same pagination system, rather opting to provide the results to be returned as a stream.

```
resp = subreddit.search(query=query, sort="new", limit=None)
```



Figure 11: Example of a post on the platform of Reddit.

Figure 11 is an example Reddit post, again the standard text and meta data is able to be seen, alongside the percentage of upvotes to downvotes.

### 3.3 Cleaning the Dataset

Throughout the investigation, there were instances where URLs and keywords of interest had been altered (deliberately or otherwise) in ways obvious to the human eye but not so easily recognised by a machine. One such example was the URL “https://bevar ukraine.dk” which was not parsed fully as one would expect due to the space in the domain name. Another example is the utilisation of alternative ways to spell keywords such as the alteration of “vaccine” to “vax”, “vaxx” and “vacc”, among others. This inevitably creates some uncertainty over the complete detection of domains and keywords of interest.

#### 3.3.1 Handling Shortened Links

While an effort was made to process shortened URLs to recover the main URL that the shortened link would redirect to, this was not possible for every URL. There are two reasons for this, firstly, there were problems processing some shortened URLs. For example, some shortened URLs would redirect to the same address infinitely or would link to another shortened URL to be expanded, and then another, and so on. Secondly, given the vast amount of URLs and the delay of receiving a response by making multiple requests to websites that may or may not be live, this process of expanding URLs is quite time-consuming. This meant that this was prohibitive on the required time scale for the amount of URLs gathered. Instead only those known domains on the list provided by the package `urlExpander` [16] were expanded. This list consists of the most well-known and popular URL shortening services, ensuring that the majority of URLs were expanded.

#### 3.3.2 Ambiguous Links

An interesting point of ambiguity was encountered when processing the CrowdTangle data. Consider the URL: “100.kr”; on one hand, one can assume that this is a mistake in attempting to write “100 kr”, referring to the amount of a hundred Swedish kroner. On the other hand, it can technically be a website registered under the South Korean domain name “kr”. This begs the question and choice whether to consider these instances as websites. To remain impartial these URLs were kept in the dataset. While

it is true that the number of ambiguous links is impossible to know without manually checking each individual entry an attempt at estimating this number was attempted. 600 different posts were manually checked over. These 600 posts corresponded to 100 for each of the platforms present in the dataset and whether or not the post was found to contain a URL. No false positives or false negatives were found.

### 3.4 Duplicate entries

Many duplicate posts were present in the dataset due to the way the dataset was compiled. For example, the same post may have been found by multiple queries, a post that contains the text “The dog and the cat” will be found both by the search “dog” and the search “cat”. If this is the case, then multiple copies of it exist in the dataset.

### 3.5 The Dataset

- **id** - Unique identifier for each post. This is a string with different formats for each of Facebook, Instagram, Twitter and Reddit.
- **time** - Time the post was created, represented as a Unix timestamp.
- **author\_id** - Post’s author’s unique identifier.
- **type\_of\_post** - E.g. Reddit posts may be a submission a post that is posted in a subreddit and the beginning of the thread, or a comment, or a reply to a submission. Posts from CrowdTangle may be a variety of different types, e.g. album, link, live\_video, etc. A Twitter post may simply be an original tweet, a retweet or a quote tweet.
- **domains** - The domains of any URLs contained in the post, presented in a comma separated list.
- **conversation\_id** - Only used for Twitter posts and is the conversation ID of the tweet, the conversation ID is always equal to the ID of the original tweet in a thread.
- **score** - Only used for Reddit posts and contains the score (combined upvote and downvote total).

- **text** - This contains the text contained in a post. If the post was made on Facebook or Instagram it also contains the text in any embedded images. If the post was a retweet or quote tweet then it also contains the text of the post it was retweeting/quoting.
- **controversiality** - Only used for Reddit posts, used to display the binary variable denoting whether a post is “controversial” or not.
- **platform** - This contains the platform that the post was made on, i.e. “reddit”, “facebook”, “twitter”, “instagram”.
- **urls** - Comma separated list of URLs.
- **user\_link** - A link to the user’s account; only used for Facebook, Instagram and Reddit posts.
- **parent\_post\_id** - If the post is a reply, quote retweet, quote tweet, etc. this field contains the original post’s ID.
- **likeCount, shareCount, loveCount, wowCount, hahaCount, sadCount, angryCount, thankfulCount, careCount, favoriteCount** - This field is only used for Facebook and Instagram posts. They contain the number of corresponding reactions to the post.
- **commentCount** - This field is only used for Facebook, Instagram and Reddit posts, it contains the number of comments a particular post has received.
- **retweet\_count, reply\_count, quote\_count, like\_count** - Only used for Twitter posts and contains the respective number of retweets/replies /quotes/likes a post has received.
- **query** - The query used to find the post. This query could consist of merely a keyword, e.g. “dog” or a boolean search e.g. “dog OR hound AND NOT cat”.
- **country** - The country the post is from, only used when a search was separated by country.
- **language** - The language the post is written in, only used when a search was separated by language.

	Raw Posts	Posts without retweets	Posts containing the term “NATO” without retweets
Dannish	1,290,035	782,415	59,450
Finnish	1,736,916	900,062	398,743
Norwegian	595,730	394,942	40,775
Swedish	2,179,531	1,075,784	456,246

Table 1: Number of collected posts in the span 31 October 2021 - 2 June 2022

Table 1 shows the number of posts that were originally downloaded, versus the number of posts that were not retweets and contained the word “NATO”. These posts were used in final analysis.

### 3.5.1 Example Data Subset

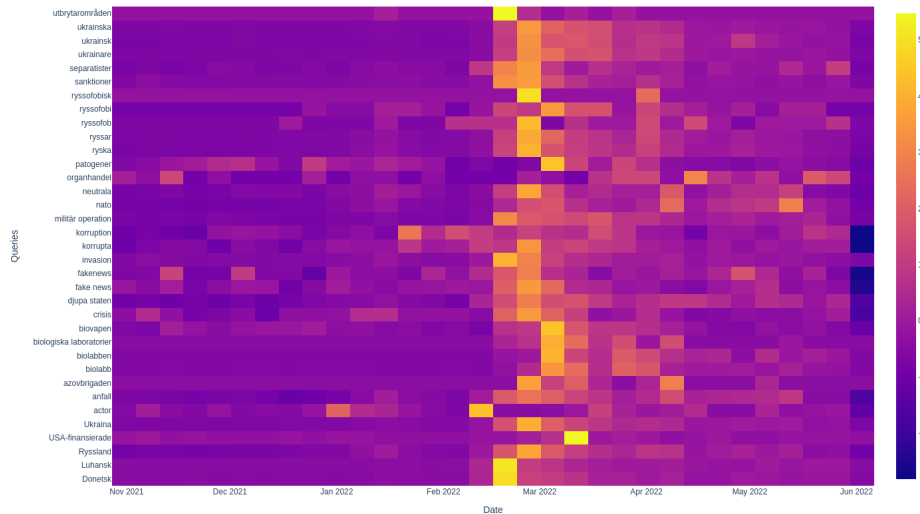


Figure 12: Heatmap of the Swedish dataset, the colour intensity is the z-score for the number of posts found in a particular week with the corresponding search

The heatmap in Fig. 12 shows the frequency of certain words and phrases of interest each week. It can be seen that their frequency rose sharply during the week the conflict began happened. While they have been sparsely used before, they rose tremendously in frequency during that brief time period.

It can be seen that the final week (far right of Fig. 12) shows less frequency compared to the other two weeks before. This is for two reasons. First, it is not capturing a full week's worth of data, as the accumulation of data did not finish before midnight on Sunday of that week. Second, the posts we were receiving were not in the form of live stream, and was, therefore, some delay to the most recent post that could be requested.

### 3.5.2 Deleted posts

It is of no surprise to anyone that has browsed the internet for long enough that posts will be deleted, either by some administrator or the original poster. This causes an issue with the analysis however, as it is impossible to analyse what is not there. To this end an attempt has been made to quantify how much information is being lost. The entire dataset was downloaded twice. Once from 31st October 2021 to 17th March 2021, then again, several months later from 31st October 2021 to 2nd June 2022. Then by observing the differences in number of posts found in the original time span we may estimate the number of posts lost. Table 2 shows the number of posts deleted in addition to the percentage of posts deleted in this time span.

	Danish	Finnish	Norwegian	Swedish
Difference	35,941	24,231	13,631	26,228
Percentage of Posts Lost	9.97	5.63	8.00	6.64

Table 2: Number of posts deleted in the span 31 October 2021 - 17 March 2022 (found when downloaded on June 5 2022)

### 3.5.3 Noise

At this point it is necessary to note that for the purposes of the analysis, there exists some noise in the dataset. For example, in one post, the word

“mål” was instead written as “määääääääääääål” to imply emphasis. The algorithms used do not understand that this consists of the same word since their character content is different and takes place neither in the beginning or the ending of the word. As such, if one was to count the occurrences of the word or classify it, the latter would not contribute to the calculation. The source of noise is the variation of word spellings. For example, the word “vaccine” being variously written as “vax”, “vaxx” and “vacc”, among others, some of which were possibly made to avoid some form of detection. The same applies to misspelled words.

## 4 Methodology

### 4.1 Data Collection Tools

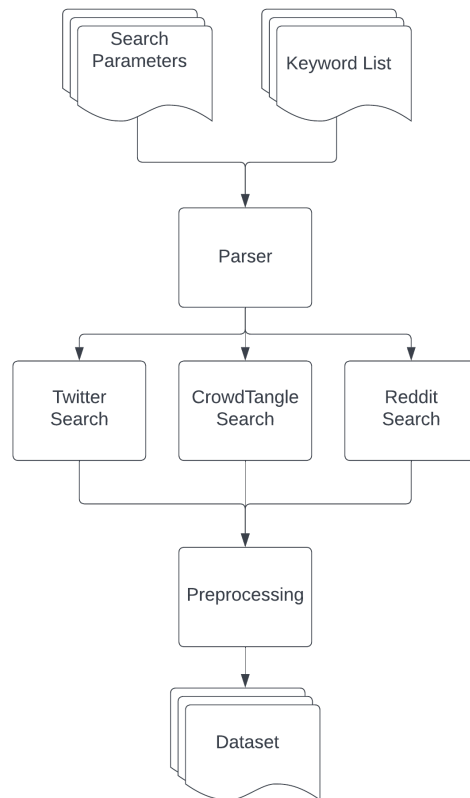


Figure 13: Data collection process

This project made use of multiple APIs from various companies to scrape information for the analysis. The search was in the form of keywords or queries, that were searched within a specific time window to the greatest extent the API would permit in each case. The overview of the data collection process is given in Figure 13. The process begins with two inputs, the search parameters and the keyword list or lists. Using these the parser creates individual searches for the platforms denoted by the search parameters. These searches are then passed to the individual platform’s API, downloading the

data before the preprocessing step. This involves extracting some elements like URLs in text, domains from URLs and zeroing out any elements not used by the platform, e.g “upvotes” for Twitter. Then finally writing the results to a CSV. Once all these searches have been iterated through, the dataset is complete.

#### **4.1.1 Twitter API**

Twitter provides a comprehensive API for research purposes. The Twitter API v2 [17] was used through the Tweepy library [18] for Python. Twitter provided an academic API key allowing us to download the necessary number of Tweets.

#### **4.1.2 Facebook and Instagram API**

For the platforms of Facebook and Instagram, the CrowdTangle API was accessed with a key provided by CrowdTangle enabling us to use the full boolean search in their API.

The “expanded\_links” field of CrowdTangle had to be filtered because invalid links were returned on multiple occasions. An example was the parsing of “https://bevar” as the website, where in the text field we found “https://bevar ukraine.dk” where the author evidently added a space character in the actual link. Another example was that of a phone number of the format “+46123456789” in said field.

#### **4.1.3 Reddit API**

Reddit provides an API that allows searches for posts (submissions) and replies to those posts (comments) in the form of a stream. For Python we found that two popular libraries exist: PRAW (The Python Reddit API Wrapper) [19] and PSAW (Python Pushshift.io API Wrapper) [20]. While PSAW provides more reliable and faster search, the Pushshift database that it queries is not updated regularly. As such, because a claim online may propagate rapidly and the fact that many posts online may be ephemeral (i.e. promptly deleted by the user or the platform), a solution that is the closest to real-time is needed, hence PRAW was chosen.

Links may be shared on a Reddit submission and/or comment that can be either embedded, or in simple plain-text. While the former is not difficult to parse, the latter is incredibly computationally expensive, so all URLs other than those embedded using the reddit embedding markup are ignored. In the example, `[shown text](www.url.com)`, `www.url.com` is found, however if in the post there was instead simply `www.url.com`, the program would not be able to find it. However, the trade-off between information loss and computational feasibility is small, as this represents a minority of links as we have seen that the vast majority of links are properly embedded.

As we have to deal with the possibility of the existence of a “title” field for submissions, but not for comments, we check for the language of both and in case they differ, we proceed with the language of the body, since it usually has more text, so that it can lead to more accurate classification. This covers the cases where, either field is empty and the cases where they both exist, but are classified as different languages. The possibility of the existence of a false positive, i.e., a submission that has a title classified correctly, and a body classified wrongly, and therefore be saved as being in the wrong language, or not saved at all if it is not one of the Nordic languages of interest, is also, consequently, low. One cause may be the fact that text on the internet does not often conform to grammatical and syntactical rules.

Reddit did not provide the search by language and/or country. To overcome this limitation the `langdetect` library [21] was used to determine the language of the post. One example is that searching for the keyword “Ukraina” may return results from many languages including the Nordic languages. It is likely that in that case the false negatives, i.e., the library being unable to classify an internet post correctly for the language it is in are fewer than the false positives. The library also provides an important function where we can more reliably classify posts with their respective language on the final CSV file, akin to Twitter and CrowdTangle.

## 4.2 Searching

Each of these tools were collated into one program. As an input we could provide either a list of keywords to search for (keyword search) or write a

more advanced boolean search. The program would then search Facebook, Instagram, Twitter and Reddit, then output the results as a CSV file. However, while all the work was done for the boolean search, in the end only the keyword search functionality was used to obtain the final results.

#### 4.2.1 Keyword Search

The keyword search is relatively straightforward, taking a text file input containing the desired keywords, the language/country and the time period to be searched.

#### 4.2.2 Boolean Search

A boolean search is more complex. The search is constructed with any of the boolean operators, i.e  $\neg$ ,  $\vee$  and  $\wedge$  may be used. A simple boolean search may be:

$$(\text{dog} \vee \text{hound} \vee \text{canine}) \wedge \neg(\text{cat})$$

However, instead of requiring the user to chain together many  $\vee$  operators together instead the user may alias a list of keywords to become a single variable. Therefore, the previous example becomes:

$$(\text{VAR1}) \wedge \neg(\text{VAR2})$$

Where VAR1 and VAR2 each point to a text file that contains the various aliases for the same word. In this example  $\text{VAR1} = \{\text{dog}, \text{hound}, \text{canine}\}$  and  $\text{VAR2} = \{\text{cat}\}$  This was originally implemented due to examining COVID data, with many people attempting to obfuscate their conversations from these very searches. For example when discussing vaccines often times rather than simply write “vaccine” or “vax” it could be “v a x” or “vaxxx”, etc. Therefore by creating a simple interface to quickly be able to add to the list of aliases for a word reduced the time required creating more complicated searches.

### 4.3 Duplicate Entries

As our dataset includes the aggregated results of all the searches that were performed, there is the possibility that same posts may have included one or more keywords or phrases of interests and consequently appear more than once and were removed from the pandas [22] dataframe before the analysis.

### 4.4 Preprocessing

A number of preprocessing steps needed to be applied before any analysis on the posts' text could be conducted. To begin with, any word with a digit in the middle of it was removed, e.g “thi8s” or “t3xt”, however not “d6” or “3”. This specific approach was selected as it didn't remove mentions of things like “3” - the telecommunications company or something like “E6”, a road. The removed “words” obviously would only add noise to the analysis, but this comes with its own downsides. By removing such words some information is inevitably lost. For example, any text written in the modified internet spelling system, commonly referred to on the internet as “leetspeak” is lost.

Twitter mentions (i.e. “@username”) were removed. This was to avoid the potential scenario where some user's username becomes over-represented in the dataset. Next, any single non-character digit, e.g “/” or “!” surrounded by whitespace was removed.

Finally, any redundant whitespace was removed; e.g “ ” became “ ”.

### 4.5 Analysis

#### 4.5.1 BERTopic

The first step of the analysis is to pass the cleaned and processed data for each language into the BERTopic algorithm. Four different BERT embedding models were used, one for each of the surveyed languages. Each model was obtained using the Hugging Face platform [23]. The Danish BERT model was created by the company Certainly (also known as BotXO) [24]. The Finnish BERT model is presented within the paper “Multilingual is not enough: BERT for Finnish”, by Antti Virtanen, et al., [25][26]. The Norwe-

gian model was created by The AI-lab at The National Library of Norway [27]. Finally the Swedish model was created by the National Library of Sweden [28][29].

#### 4.5.2 Stopword Removal

With the embeddings for each post the post’s stopwords are removed to create better topic representations. The stop word list used was the spaCy package for Python [30]. The issue here is that the stopword lists contained do not account for any regional dialect differences in the languages. As an example, Norway has two major dialects, Bokmål and Nynorsk. Bokmål is the more popular of the two, however if even a very small percentage of the dataset is in Nynorsk and contains some stopword that isn’t in Bokmål then that stopword will likely become over represented in the topics for Norwegian.

#### 4.5.3 Topic Reduction

When generating topics the algorithm will just generate as many topics as clusters that are found by HDBScan. To reduce this number (often in the thousands), starting with the smallest topic, topics that share a cosine similarity of greater than 0.915 are combined, then their topic representation and c-TF-IDF score are recalculated until there were no more topics that were “similar”.

However this is often insufficient and there may still be dozens of apparently different topics. To reduce this number further, the dendrogram of the topics was plotted and then those topics that could be logically combined were, until a suitable number of topics for analysis (generally less than 10) was attained.

#### 4.5.4 Topics Over Time

Finally the c-TF-IDF scores were recalculated, taking the timestamps of the posts into account, using a number of bins approximately equal to the number of weeks the data has been collected over, in this case 30. Then the original topics were plotted showing their prevalence over time and the new

weekly topic representations are printed out, leading the way for others to perform further analysis.

## 5 Results

### 5.1 Raw Topics Before Topic Reduction

Each of the language’s datasets’ posts having been run through the BERTopic algorithm with their corresponding BERT model produce varying amounts of topics.

Language	Number of topics	Number of Posts
Danish	233	45,371
Finnish	1,121	210,208
Norwegian	209	31,081
Swedish	943	238,152

Table 3: Topic counts before topic reduction

As expected and seen in Table 3 there seems to be less division in discussion around NATO in Denmark and Norway. This makes sense as both of these countries are already member states and have been since 1949. However with Sweden and Finland the conversation is much more varied with the currently ongoing conflict in Ukraine and membership is a contemporary political issue.

Figures 14 - 17 contain the topics generated by the BERTopic algorithm for the respective dataset mapped onto 2-dimensions. Each topic is represented by a colour and each point on the graph is an individual document. Each point has some level of transparency, therefore more intense colouring implies a more definite cluster. The name of each topic is merely a number (denoting the relative ranking of the topic in terms of size) and the top 3 words as calculated by the c-TF-IDF score.

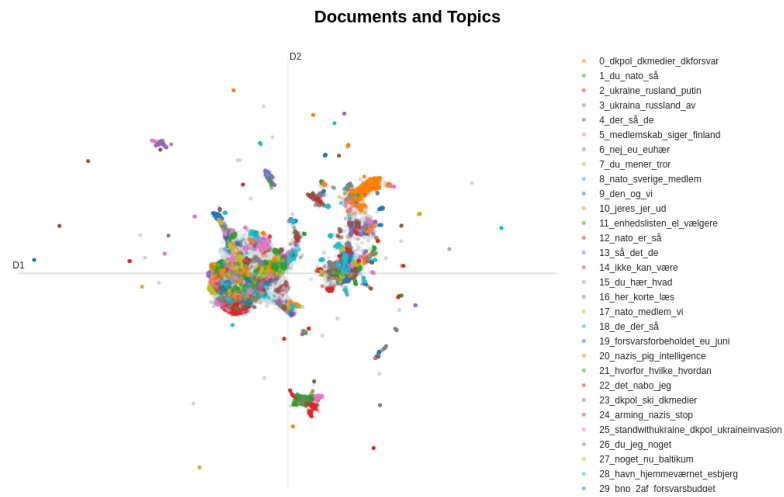


Figure 14: Clusters generated by BERTopic from the Danish dataset, mapped onto 2-dimensions by UMAP

Fig. 14 and Fig. 16 for the Danish and Norwegian datasets respectively, seem to imply that there are somewhat distinct groups within the generated topics. There are multiple isolated and semi-isolated groups of points. This is in contrast to the clustering in Fig. 15 and Fig. 17 representing the Finnish and Swedish datasets. The mapping that UMAP has produced seems to show that there are a large number of documents that are disparate and are not as easily separable.

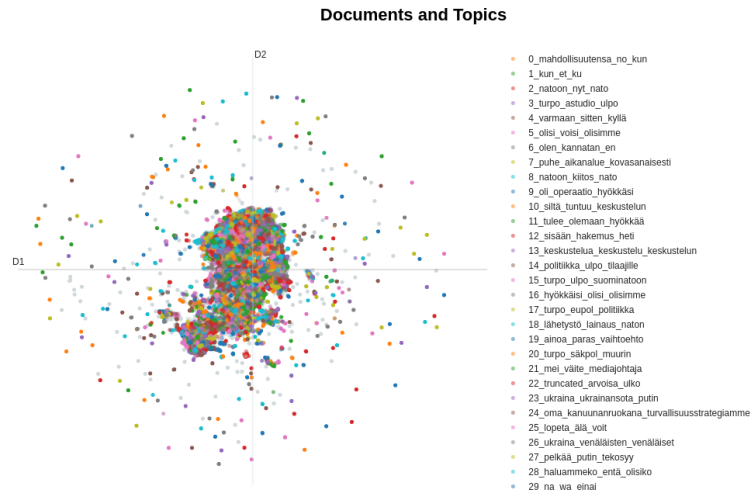


Figure 15: Clusters generated by BERTopic from the Finnish dataset, mapped onto 2-dimensions by UMAP

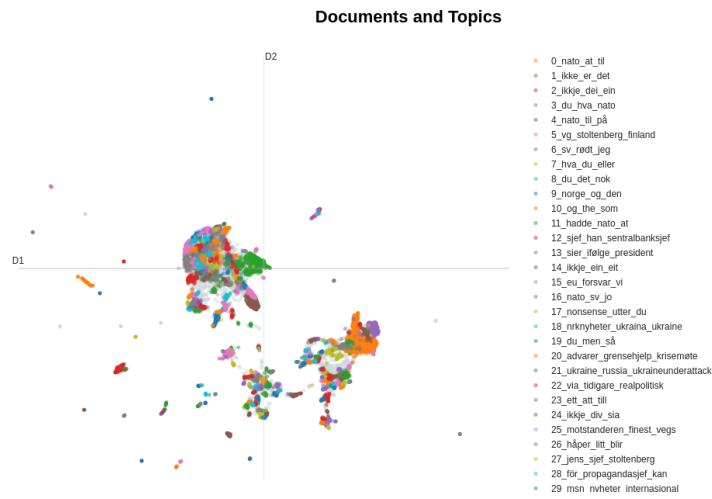


Figure 16: Clusters generated by BERTopic from the Norwegian dataset, mapped onto 2-dimensions by UMAP

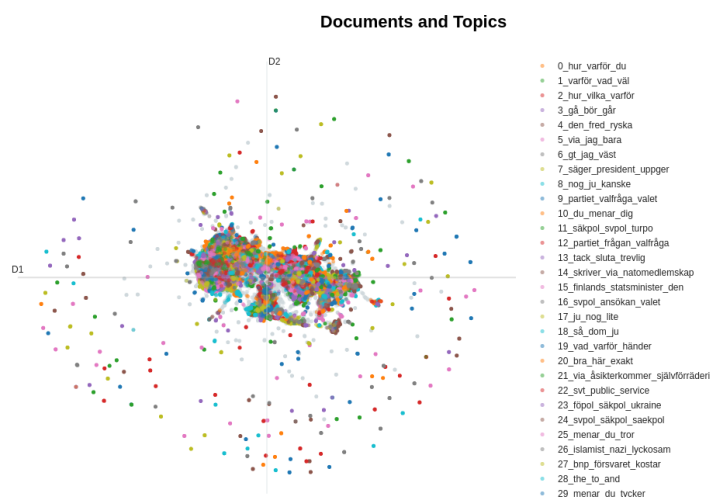


Figure 17: Clusters generated by BERTopic from the Swedish dataset, mapped onto 2-dimensions by UMAP

## 5.2 Topic Reduction

Language	Number of Topics
Danish	50
Finnish	40
Norwegian	9
Swedish	858

Table 4: Topic count after automatic topic reduction

The topics having been reduced by combining all those that have a cosine similarity score greater than 0.915 reveals that the overwhelming majority of topics were in fact, very similar. The only exception to this is the Swedish model. This could be due to a number of reasons, but it is likely that due to the non-deterministic nature of UMAP and potentially of HDBScan and the combination of algorithms simply created a poor representation of the data into lower dimensions. It may also be the case however that the problem is poor choice of hyperparameters.

Further investigation into the Swedish dataset shows that the topics are too disparate to combine. It would of course be possible to run the algorithm again to create new topics. However, given the amount of time required to rerun the algorithm this is as far as the Swedish dataset will be analysed.

Even with this heavily reduced number of topics any analysis would be far too granular to be comprehensible. Each topic in and of itself will have many sub-topics, one for each week over the examined time period. More reduction is required for both the Danish and Finnish datasets.

Note, that as the UMAP algorithm is non-deterministic, the projections produced for the topic reductions appear slightly different.

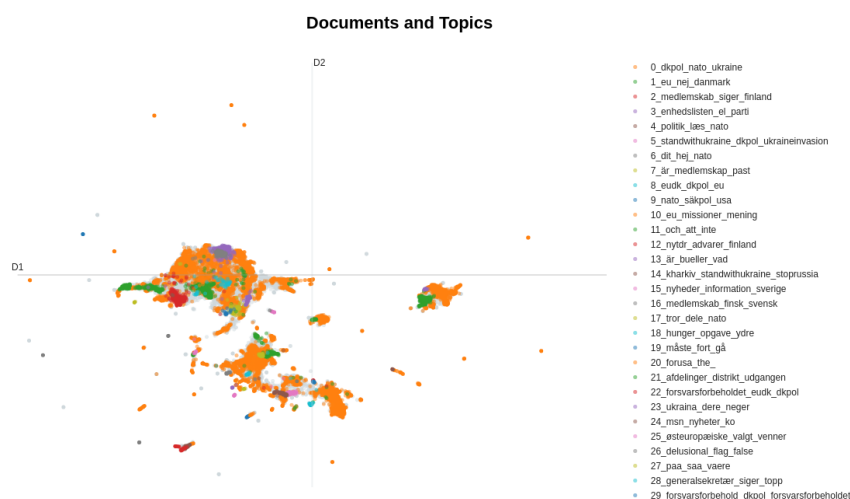


Figure 18: Clusters generated by the Danish dataset by BERTopic and having topic reduction applied, mapped onto 2-dimensions by UMAP

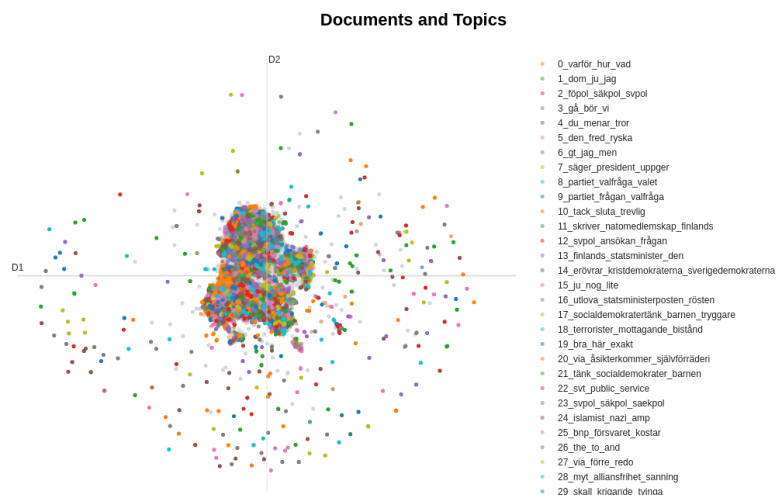


Figure 19: Clusters generated by BERTopic and having topic reduction applied, from the Swedish dataset, mapped onto 2-dimensions by UMAP

### 5.3 Further Topic Reduction

To make the topics possible to be analysed, the number of topics again needed to be reduced. To this end, the dendrogram of the clusters was examined for the Danish and Finnish dataset, with the Norwegian dataset already has a number of topics that can be easily parsed by a human.

Examining the dendrogram in Figure 20 for Danish and Figure 21 for the Finnish topics, the decision was made to cut the dendrograms at approximately 1.7 and approximately 1.55 respectively as this was deemed a sufficiently low enough distance to produce results to be legible to a human (in the range of 5-15 topics).

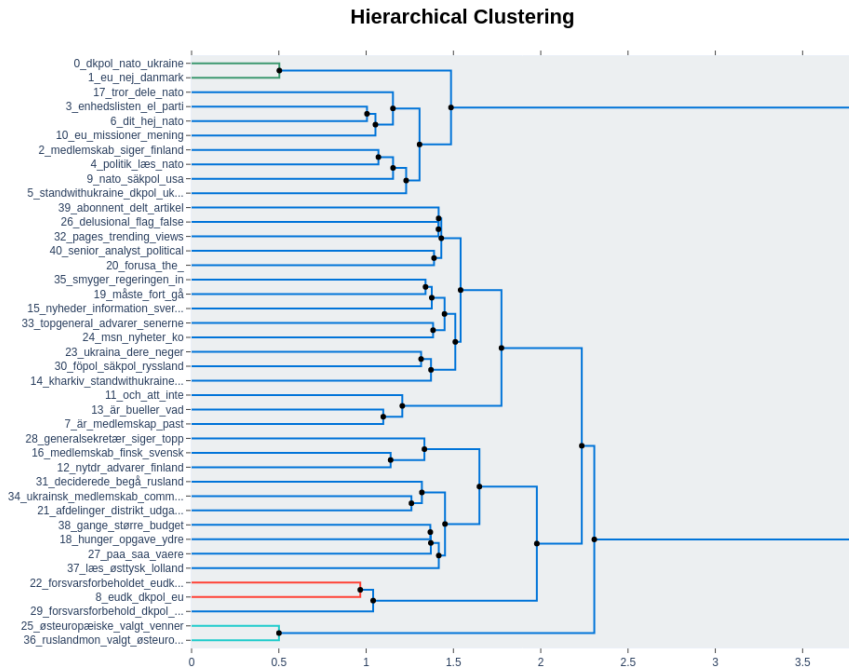


Figure 20: Hierarchical Topic clustering of the Danish dataset after topic reduction has taken place

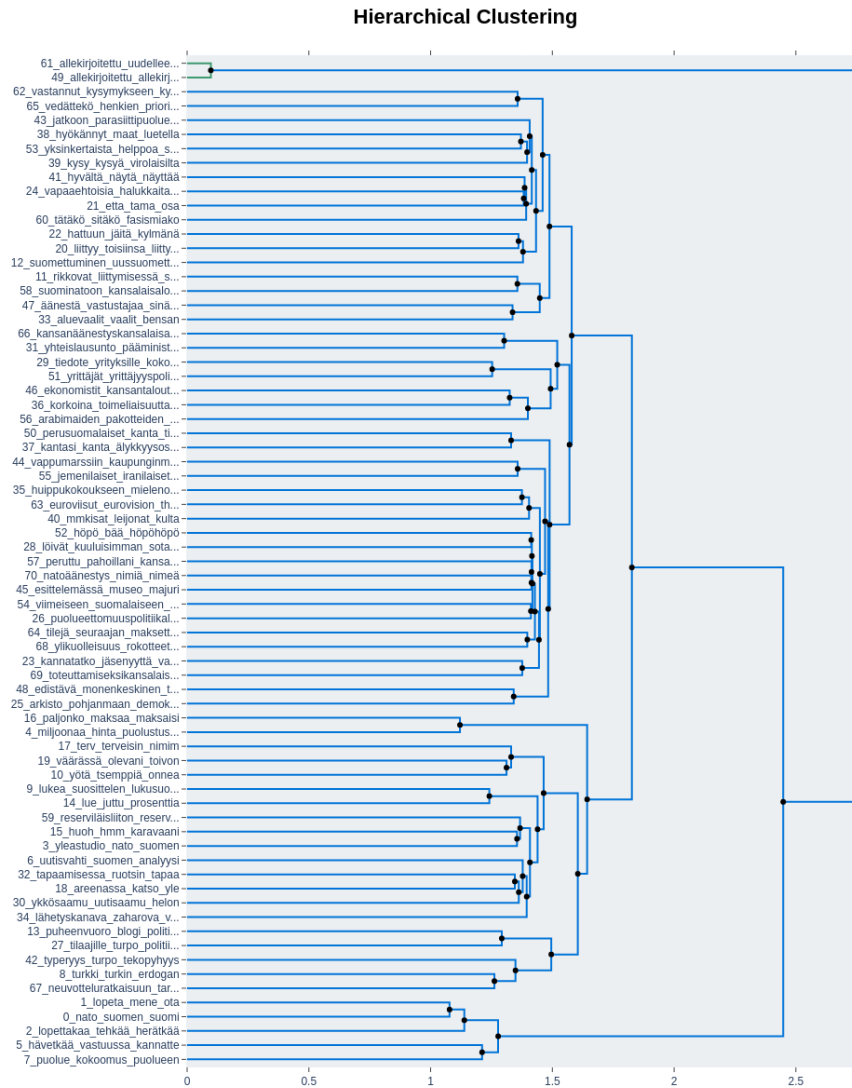


Figure 21: Hierarchical Topic clustering of the Finnish dataset after topic reduction has taken place

Language	Number of topics
Danish	6
Finnish	8
Norwegian	9

Table 5: Number of topics following automatic and manual topic reduction processes

#### 5.4 Topics Over Time

In both the Danish and Norwegian datasets the conversations around NATO seemed to pick in late May, at the start of the conflict, as can be seen in Fig. 22 and Fig. 24. However, for the Finnish dataset in 23 the discussion around NATO membership seems to peak towards the end of May. This lines up with Finland’s ongoing (as at the time of writing) application to join NATO.

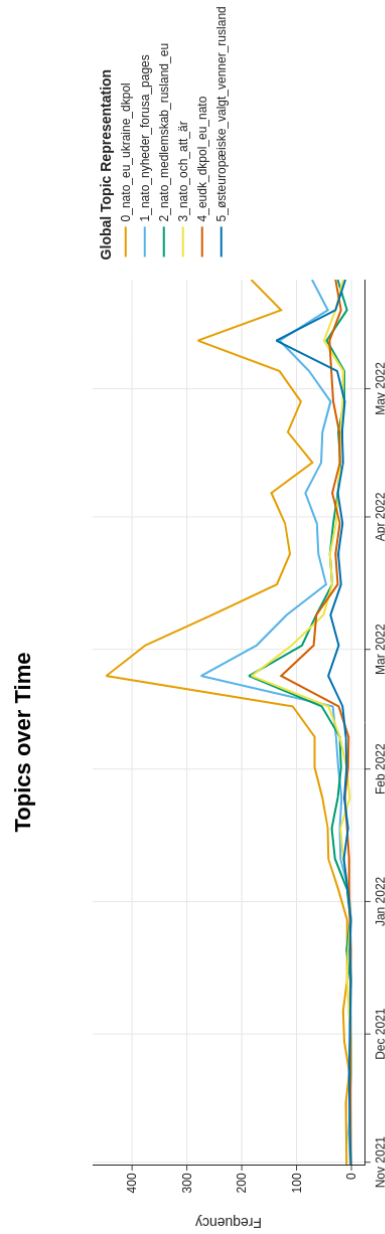


Figure 22: Temporal frequency of the top 10 topics for the Danish dataset

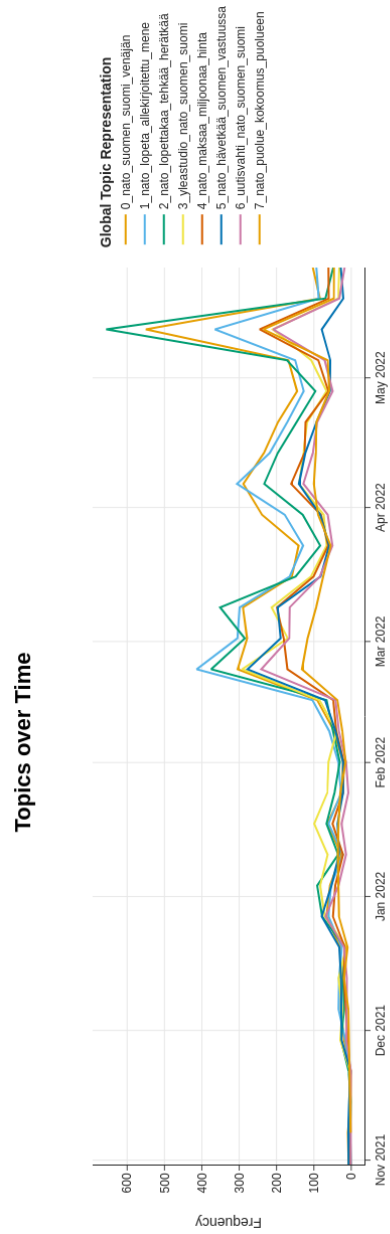


Figure 23: Temporal frequency of the top 10 topics for the Finnish dataset

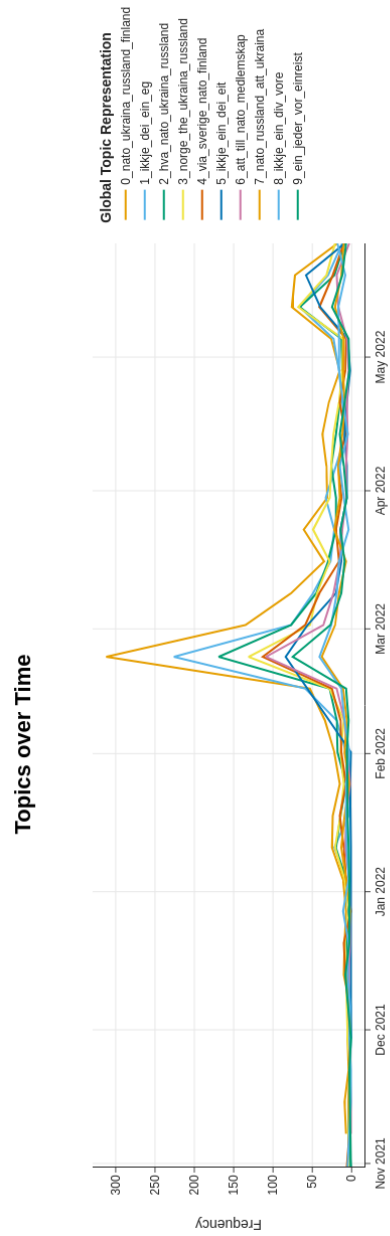


Figure 24: Temporal frequency of the top 10 topics for the Norwegian dataset

	Word 1	Word 2	Word 3	Word 4	Word 5
Week 0	karhutilalle	ydintuhoa	keskuksessa	kantamassa	laukkuja
Week 1	mykkäsen	mallioppilastaan	aitahan	kylttejen	polskit
Week 2	käsikranaatin	kapasiteettia	hoitokuormituksen	sairaanhoidon	haaveillaan
Week 3	ismay	germans	purpose	sg	americans
Week 4	etupiiripuheellaan	myllyyn	hyökkäysoperaatioitalibya	operaatiokäyttöön	saabia
Week 5	suuttui	oh	rintamakarkureita	kirjoitushommat	turskan
Week 6	txf	myyntiaika	neliön	puolustusteollisuusyhdistyksen	mukaannauraen
Week 7	sängynkin	hallituspuolueissamme	savuttava	aivopes	copycat
Week 8	kalenterimerkintä	lippaat	kaliberi	huulilla	odotukset
Week 9	aukiehkä	aivankuten	paikannusta	öyhötiöyhöti	mikrosirun
Week 10	tapaamisia	huolisikohan	hintakattosoppari	seurakuntavaaleissa	erkitkään
Week 11	selkeyttä	jengiinniettikää	venetsueelaan	liittyäei	suomenvatulointia
Week 12	lumilinko	lepertelypuheluita	remmit	koronarokotepassiopio	hautajaisia
Week 13	henkilöjäsenyyksiä	propagandakolumnistien	varasi	värittäämään	usuttamina
Week 14	tellusta	kylmänsodan	feminismi	nimettömien	pakkodemokratiaa
Week 15	vastustaisi	kärnän	vatuloi	liitto	kapinallismaakuntien
Week 16	meidæn	pöksyssä	lt	run	huikkii
Week 17	pamu	tiiseri	mielentilan	otto	koomikko
Week 18	sököringissä	vtun	natot	opittua	hyökkäs
Week 19	viuh	ykkösiä	huijaus	kehua	pele
Week 20	vahvenmalle	tok	mahdollisuutensa	tik	hakuprosessin
Week 21	mahdollisuutensa	pääsemme	menettää	hyökätä	hauskin
Week 22	golgatalle	hiottava	erkin	salaliitossa	tietolähde
Week 23	kuvioista	maanpetteriä	sunlaiset	junttipäisesti	peittelyyn
Week 24	injektiolla	nysväämistä	sönkkäämistä	pillerin	iik
Week 25	ylläri	retoriikalla	marko	jahtaava	näkstä
Week 26	reenattu	tuoppi	myönteisen	jouluksi	korea
Week 27	miekkarissa	vassarien	harmittaa	juhla	just
Week 28	hyötyy	ultra	vouhkaava	kaulailevan	apunoita
Week 29	natottajien	maailmanliitto	väestökontrollin	kaikile	murphyn

Table 6: Weekly top 5 words for topic 0 in the Finnish dataset.

Examining Table 6 it becomes apparent that interpreting this data is not a trivial exercise. There is 30 weeks worth of data, with 5 words in each and this is only for one topic.

Immediately the topics representations for the Norwegian dataset draw attention. Words like “ikkje”, “dei” and “att” make up the topics. This could be in part due to some dialect issue as the Spacy stopwords list only covers

Bokmål, not Nynorsk. It may also be due to posts being incorrectly tagged as Norwegian when they were in fact Danish or Swedish. It is difficult to tell exactly what the problem is without the requisite linguistics background.

There are some interesting results, for example, the word “ydintuhoa” (nuclear destruction) in week 0 versus “maailmanliitto” (world alliance) in week 29. As I cannot speak Danish, Finnish or Norwegian, analysis is somewhat difficult in this area. A professional with expertise in the language would be best suited to analyse these topics to determine if they are meaningful in any way.

## 6 Conclusions

This project had many limitations and many future avenues of exploration. A streamlined data collection system would increase the speed of the process. As always, adding more data from different platforms would allow for a more comprehensive examination of online discourse. There are also multiple obvious paths forward for this project. As it stands, the method used in the analysis was unoptimized. However, while optimizing the BERTopic algorithm would be the logical first step to take, this would require some labelled data to have some level of “base truth” to compare the differing models generated.

There are however, many limitations to the project. Each platform has its own quirks and limitations on what data may be sourced. Then there is also a myriad of caveats to the analysis itself. As the the BERT models used were not language agnostic. If a different language or even dialect is used in a post, the model may not create accurate an accurate embedding. One of the largest limitations however, is the lack of optimization in the algorithm. It is known that the hyperparameters of the algorithm do have a good level of impact on the final results, however without some way to compare models and the resources to optimize, it isn't possible to address this issue.

The results here are a good first step for further analysis, this could take

many different forms, from qualitatively assessing the results presented here, or improving on the method performed here.

## 6.1 Limitations and Caveats

### 6.1.1 Facebook/Instagram

CrowdTangle only provides access to posts made by public-facing, non-personal accounts. In addition to this it does not allow access to comments made on posts. This means that we are unable to model the spread of claims amongst the general populace of the platforms.

### 6.1.2 Reddit

The Reddit search API only provides the most recent 1000 posts corresponding to the search parameters [31], this means that we are unable to find all posts that relate to the claim of interest if the claim is sufficiently old and/or popular.

In 2014 Reddit removed the ability to see the individual upvote and downvote count, instead combining these into one aggregate score [32]. This means that we are unable to accurately gauge interaction with a post. For example, let's say that a post has received 100 upvotes and 100 downvotes, the aggregate score is merely 0. Reddit does compensate this by providing a binary **controversiality** variable that tags posts with a high number of upvotes and downvotes, however this is not enough to accurately estimate the engagement.

Reddit also stems the input of the search, so, for example, the search may be “material OR matrix” which could then be stemmed to “mat OR mat” [33]. This can create a large amount of false positives in the search results. The false positives themselves are not inherently a problem as they have been dealt with, however given the already limited amount of search results coupled with this further reduction means that the probability that we find all posts relating to any specific claim of interest is lowered significantly.

Another issue is that we are unable to search text on images on Reddit.

This means that we miss any screenshots of tweets or Facebook posts for example. We also miss any relevant image macros that may be posted.

### **6.1.3 Twitter**

Twitter is generally quite free of limitations. However, the sole sore point is the rate limit for pulling replies and the users who have liked a tweet, two metrics that correspond to spread. Whereas pulling hundreds or thousands of tweets in the span of a few seconds, finding the users that liked those posts can take an order of magnitude longer. To this end, it was not entirely feasible to do this for each claim investigated.

### **6.1.4 Data Collection**

There are some inherent limitations when pulling the data from the sources due to the nature of the different languages. Finnish agglutination is a problem, capturing just one “word” when that word may be present and changed, agglutinated into some other form is incredibly difficult. Take for example the word “istua” (to sit down) it may appear as “istahtaa” (to sit down for a while) or “istahtaisinkohan” [34], in addition to a myriad of other forms.

### **6.1.5 BERT**

There are two main limitations of BERT. Firstly, the models used were language specific and, more importantly, dialect specific. Firstly, if the model wasn’t trained using a specific dialect then it may struggle to create accurate embeddings. Secondly, BERT is unable to capture the meaning behind image macros, or so called “memes”. As generally these memes use some static image to convey meaning to an otherwise insignificant piece of text.

### **6.1.6 BERTopic**

BERTopic uses two different algorithms; UMAP and HDBScan. Both of these algorithms have parameters that may be tuned, indeed it is known

that the results UMAP provide are heavily influenced by modifying its parameters. This implies the highly likely event that the dimension reduction performed here was sub-optimal. Similarly with HDBScan, choosing the number of minimum points in a cluster is of huge importance, given that in this analysis the goal is to be able to find some fine-grain distinction between different conversations regarding the same topic. However, there is no method for finding these values other than a grid search, which is resource intensive.

The amount of topics generated by this type of analysis in conjunction with the dataset used was an issue. With such a large number of topics being produced, even after automatic topic reduction this implies that there are more distinct conversations taking place than used in the analysis. There is of course the question if these topics are actually distinct conversations taking place or merely the same discussion taking place using different verbiage. The choice to combine topics was made as a concession to legibility and so these topics may not be as precise and clean-cut as desired.

## **6.2 Future Work**

### **6.2.1 Data Collection**

The data collection tools could be updated and retooled. As it stands, they are quite simplistic, optimised and missing features. Simultaneously querying all platforms concurrently would significantly speed up the collection process. Time constraints precluded this modification to the process. Additionally it would be potentially useful to develop similar tools for other platforms, e.g Telegram and YouTube.

### **6.2.2 Dataset**

While having more recent data would be of great benefit, the 7-month dataset is quite extensive. However, the conspicuous lack of Reddit data is an issue, as is the lack of data from other, unavailable platforms.

As mentioned, often posts had been deleted by the time the data gathering had begun. While it is unknowable how many of these posts were deleted by

a platform’s moderating staff and/or algorithm, it is some non-zero amount. Having these posts included may lead to new information.

### **6.2.3 NLP techniques**

It is possible to create topics with BERTopic trained on data with a word n-gram size greater than 1. In other words, it’s possible to train the model to pair words together, for example the words “Slava” and “Ukraini” or “prime” and “minister” may appear so often together as to warrant their own position in a topic as one word/phrase. However this method may create mixed results. All of Danish, Finnish, Norwegian and Swedish construct compound words quite regularly, so the idea of multiple words appearing together consistently and providing additional meaning may not be realistic.

### **6.2.4 Optimizing the Algorithm**

Assuming that there was some effort to label some of the data, optimization may be a distinct possibility to improve the results presented here. The algorithm used in the analysis was merely the standard, unoptimized algorithm. There are a few different ways this analysis could be improved. The clustering algorithm, HDBSCAN is the main bottleneck when it comes to performance other than generating the embeddings. To this end, switching this algorithm out with another, faster algorithm may allow for parameter optimization for the UMAP algorithm, generating better results. However, it is also possible to optimize without this change of algorithm, but as previously stated, this is computationally expensive.

### **6.2.5 Semi-Supervised Topic Modelling**

A quick sketch of the semi-supervised method is provided here for context. Rather than having data with no target labels, instead assume that there exists some part of the dataset that have known labels. When the UMAP algorithm clusters the embeddings it already knows some (not all) of the datapoints that should be clustered together, influencing how the rest of the unknown data is clustered.

It is possible that the data is much more amenable to analysis with a different approach such as this. The issue arises with providing these pre-supposed

labels. If some topics were provided by some experts who had already gathered, collated and then generated these topics based off some method of theirs then this method may well be appropriate. However, in this analysis no such premade topics were able to be found and furthermore the problem of tagging a sufficient number of posts with each topic would be very time consuming and difficult.

### **6.2.6 Further Investigation Into Results**

As it stands there is a lot of knowledge to be gleaned from the results in and of themselves, both regarding the methods used and the results unto themselves. As it stands, without any advanced linguistics capability more detailed interrogation of the data is difficult. Having an expert with domain knowledge examine the generated topics for each of Danish, Finnish and Norwegian to discover any insights if they exist within the results.

Another interesting avenue of exploration would be to find collections of specific pieces of propaganda, then examine if these stories impacted the conversation by lining up the time that these stories were published with the topics.

## **6.3 Ethical Implications**

The data obtained contained the user's ID for the respective platform, thereby providing a clear line of identification to the account of the user. This could be an issue if the data was processed in any way that examined individual users' posts. However, the analysis performed here does not examine any of the posts individually, rather it only examines them as a totality.

It is exceedingly likely that the average user didn't know that they were consenting for their posts to be used in such an analysis. The real question is in whether or not they would consent to such a use. Even if they have technically agreed to the platforms' rules the legality of the situation is divorced from the ethical question. Having their conversation surveilled and collated violates the societal norm of the expectation of privacy, even in a public space like these platforms. The issue is not that the data is being

used, rather, what it is being used for. In this case the poster is unlikely to agree the use of their post if it includes tracking and profiling to quash political dissidents. However, they are far more likely to agree to some use case where their data is being used for a less nefarious purpose.

Nevertheless, using the Contextual Integrity language developed by Nissenbaum [35], the flow of data here appears thus;

- Subject: Platform users
- Sender: Platform
- Recipient: Researcher
- Information type: posts, account identifiers
- Transmission principle: Approval of research request from Platform (in the case of CrowdTangle and Twitter)

The usual ethical problem is that of the transmission principle, that is, how the data was obtained from the source. If the data has been taken “without a warrant” so to speak then this becomes more of an ethical problem. As it stands there is a clear line of accountability for the data, with both CrowdTangle and Twitter providing explicit access (by way of providing API access keys after applications) to this data. There is a discussion to be had about whether or not this permission constitutes a sufficient constraint on the flow of the data, however that is beyond the scope of this paper.

## References

- [1] D. M. Blei and J. D. Lafferty, “Dynamic topic models,” in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 113–120. [Online]. Available: <https://doi.org/10.1145/1143844.1143859>
- [2] M. Grootendorst, “Bertopic: Neural topic modeling with a class-based tf-idf procedure,” 2022. [Online]. Available: <https://arxiv.org/abs/2203.05794>

- [3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” 2018. [Online]. Available: <https://arxiv.org/abs/1810.04805>
- [4] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2013. [Online]. Available: <http://arxiv.org/abs/1301.3781>
- [5] L. McInnes and J. Healy, “Umap: Uniform manifold approximation and projection for dimension reduction,” *ArXiv*, vol. abs/1802.03426, 2018.
- [6] A. P. Andy Coenen, “Understanding umap,” <https://pair-code.github.io/understanding-umap/> [Accessed: 08/09/2022], 2022.
- [7] L. Deng, “The mnist database of handwritten digit images for machine learning research,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.
- [8] L. McInnes, “Performance comparison of dimension reduction implementations,” <https://umap-learn.readthedocs.io/en/latest/performance.html#performance-scaling-by-dataset-size> [Accessed: 06/08/2022], 2018.
- [9] L. McInnes, J. Healy, and S. Astels, “hdbscan: Hierarchical density based clustering,” *J. Open Source Softw.*, vol. 2, p. 205, 2017.
- [10] S. A. Leland McInnes, John Healy, “How hdbscan works,” [https://hdbscan.readthedocs.io/en/latest/how\\_hdbscan\\_works.html](https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html) [Accessed: 01/08/2022], 2016.
- [11] M. Grootendorst, “Bertopic api documentation,” <https://maartengr.github.io/BERTopic/api/bertopic.html> [Accessed: 24/08/2022], 2022.
- [12] J. Novembre, “Pritchard, Stephens, and Donnelly on Population Structure,” *Genetics*, vol. 204, no. 2, pp. 391–393, 10 2016. [Online]. Available: <https://doi.org/10.1534/genetics.116.195164>

- [13] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *the Journal of machine Learning research*, vol. 3, pp. 993–1022, 2003.
- [14] A. Bricken, “Does bert need clean data? part 2 - classification.” [https://bricken.co/nlp\\_disaster\\_tweets\\_2/](https://bricken.co/nlp_disaster_tweets_2/) [Accessed: 17/04/2022], Nov 2021.
- [15] “Ranking of social networks in sweden as of may 2021, by market share,” <https://www.statista.com/statistics/621353/most-popular-social-networks-in-sweden-by-page-views/> [Accessed: 14/03/2022].
- [16] L. Yin and M. Brown, “Smappnyu/urlexpander: v0.0.35,” Dec. 2018. [Online]. Available: <https://doi.org/10.5281/zenodo.2458546>
- [17] “Twitter api,” <https://developer.twitter.com/en/docs/twitter-api> [Accessed: 05/02/2022].
- [18] J. Roesslein, “Tweepy documentation — tweepy 4.6.0 documentation,” <https://docs.tweepy.org/en/stable/> [Accessed: 02/03/2022], 2022.
- [19] B. Boe, “Praw: The python reddit api wrapper — praw 7.5.0 documentation,” <https://praw.readthedocs.io/en/stable/> [Accessed: 04/04/2022], 2021.
- [20] D. Marx, “Psaw: Python pushshift.io api wrapper (for comment/submission search) — psaw 0.0.12 documentation,” <https://psaw.readthedocs.io/en/latest/> [Accessed: 12/04/2022], 2020.
- [21] M. Danilák, “Python package index - pypi — langdetect,” <https://pypi.org/project/langdetect/> [Accessed: 01/04/2022], 07/05/2021.
- [22] W. McKinney *et al.*, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.
- [23] “Rate limits,” <https://huggingface.co/> [Accessed: 02/05/2022], 2022.
- [24] M. H.-B. Certainly/BotXO, “Danish bert,” <https://huggingface.co/Maltehb/danish-bert-botxo> [Accessed: 01/05/2022], nov 2021.
- [25] P. v. P. TurkuNLP, “Finnish bert,” <https://huggingface.co/TurkuNLP/bert-base-finnish-cased-v1> [Accessed: 01/05/2022], may 2021.

- [26] A. Virtanen, J. Kanerva, R. Ilo, J. Luoma, J. Luotolahti, T. Salakoski, F. Ginter, and S. Pyysalo, “Multilingual is not enough: Bert for finnish,” 2019. [Online]. Available: <https://arxiv.org/abs/1912.07076>
- [27] P. NbAiLab, “Norwegian bert,” <https://huggingface.co/NbAiLab/notram-bert-norwegian-cased-080321> [Accessed: 01/05/2022], feb 2022.
- [28] P. v. P. KBLab, “Swedish bert,” <https://huggingface.co/KB/bert-base-swedish-cased> [Accessed: 01/05/2022], may 2021.
- [29] M. Malmsten, L. Börjesson, and C. Haffenden, “Playing with words at the national library of sweden – making a swedish bert,” 2020. [Online]. Available: <https://arxiv.org/abs/2007.01658>
- [30] Explosion, “spacy,” <https://spacy.io/> [Accessed: 22/08/2022], 2018.
- [31] B. Boe, “Praw 3.6.2 documentation,” [https://praw.readthedocs.io/en/v3.6.2/pages/getting\\_started.html](https://praw.readthedocs.io/en/v3.6.2/pages/getting_started.html) [Accessed: 27/03/2022], 2014.
- [32] Deimorz, “Reddit changes: individual vote counts no longer visible,” <https://www.reddit.com/r/announcements/comments/28hjga> [Accessed: 27/03/2022], 2014.
- [33] “Reddit search,” <https://www.reddit.com/wiki/search> [Accessed: 27/03/2022], 2017.
- [34] M. Talks, “Rich vocabulary,” [http://mttalks.ufal.ms.mff.cuni.cz/index.php?title=Rich\\_Vocabulary](http://mttalks.ufal.ms.mff.cuni.cz/index.php?title=Rich_Vocabulary) [Accessed: 24/08/2022], 2015.
- [35] H. Nissenbaum, *Privacy in Context: Technology, Policy, and the Integrity of Social Life*. Stanford University Press, 2009. [Online]. Available: <https://books.google.se/books?id=y4FOswEACAAJ>

## 7 Appendix

### Keyword Lists

**Danish keywords:** militær operation, Afnazificering, nazister, folkeret, folkeretten, invadere invasion, Kharkiv, Kiev, Kyiv, Lavrov, militær, militæroperation, Moskva, Nato, nazi, Norden, nordiske lande, Putin, russisk,

Rusland, sanktion, sanktionerne, sanktioner, Ukraine, ukrainsk, ukrainere, Zelensky, fædrelandet, Kherson, krig, atomkrig, atomvåben, arsenal, fred, våben, oligark, Navalnyj, folkeret, general, oberst, major, forsker, besætte, besættelse, besat, Danmark, Norge, Sverige, Finland, Grønland, Færøerne, russiske ambassade, russiske ambassadør

**Finnish keywords:** ukraina, ukrainansota, nato, venäjä, pohjoismaat, sota, propaganda, pakotteet, pakolaiset, ukraina, Ukrainan sota, nato, venäjä, pohjoismaat, sota, sodan, propaganda, pakotteet, pakotteita, pakolaiset, pakolaisia, pakolaisista, Euroopan unioni, EU

**Norwegian keywords:** militær operasjon, angrep, angripe, arsenal, atomkrig, atomvåpen, basepolitikk, bombe, denazifisering, fedrelandet, folkerett, folkeretten, forsker, fred, general, invadere, invadert, invasjon, Kharkiv, Kherson, Kiev, konflikt, krig, Kyiv, Lavrov, major, militær, militærop-  
erasjon, missil, Moskva, Nato, Navalnyj, nazi, nordområde, nordområdene, nynazist, oberst, okkupant, okkupere, oligark, Putin, rakett, russisk, Russland, sanksjon, sanksjoner, Stoltenberg, Ukraina, ukrainere, ukrainsk, våpen, verdenskrig, WW III, Zelensky

**Swedish keywords:** militär operation, krig denazifiera, azovbrigaden, invasion, biovapen, biolabb, biolabben, biologiska laboratorier, ryska, Ryssland, Ukraina, sanktioner, ukrainsk, ukrainska, ukrainare, ryssar, ryssofob, ryssofobi, ryssofobisk, neutrala, korrupta, korruption, separatister, utbrytarområden, Donetsk, Luhansk, anfall, djupa staten, organhandel, fake news, fak-  
enews, crisis, actor, patogener, USA-finansierade

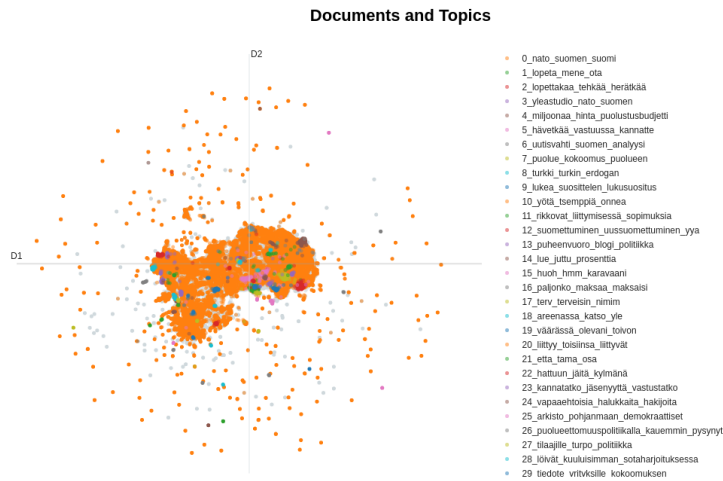


Figure 25: Clusters generated by the Finnish dataset by BERTopic and having topic reduction applied, mapped onto 2-dimensions by UMAP

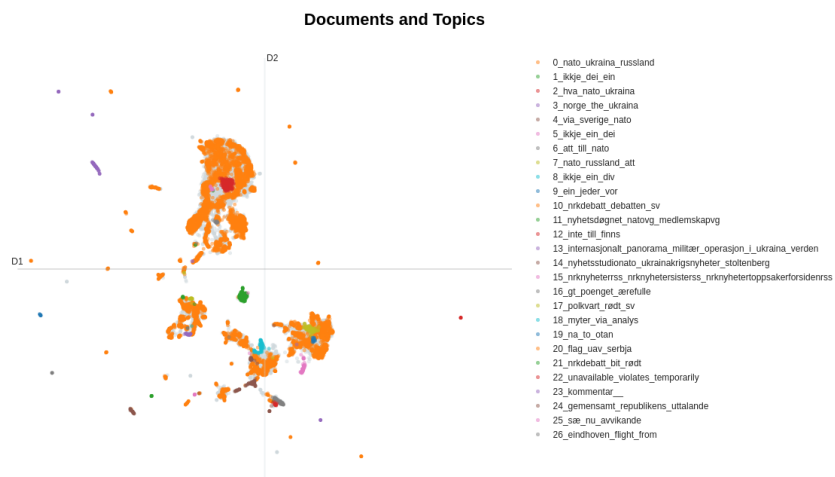


Figure 26: Clusters generated by the Norwegian dataset by BERTopic and having topic reduction applied, mapped onto 2-dimensions by UMAP

	Word 1	Word 2	Word 3	Word 4	Word 5
Week 0	forsvarsforlig	dkpol	statsministeren	dkforsvar	fröderberg
Week 1	belarus	lukasjenko	refugeecrisis	drama	eupol
Week 2	migrantkrise	dkforsvar	dkpol	gymnasieelever	brigadens
Week 3	ureflekterende	eftersnakkere	regeringernes	indsoldater	deret
Week 4	dkpol	dkforsvar	fru	dkmedier	brookings
Week 5	dkforsvar	dkpol	hegnene	innovationsfond	forarbejdet
Week 6	reznikov	statslige	dkpol	trine	alvorlige
Week 7	revanchisme	dkpol	aske	fe	risikovurdering
Week 8	nytaarstale	undsætning	dkpol	lænker	kriseberedskabsoperationer
Week 9	dkpol	dkmedier	verdenifolgegram	dkforsvar	uacceptabelt
Week 10	dkpol	dkmedier	dkforsvar	støvring	foråret
Week 11	dkpol	dkmedier	dkforsvar	opstillettropper	nabolande
Week 12	dkpol	dkforsvar	dkmedier	udenrigs	krav
Week 13	dkpol	mexmur	dkforsvar	dkmedier	regeringen
Week 14	dkpol	dkforsvar	dkmedier	usa	thefloatingtemplebyextraterrestrialtech
Week 15	dkpol	dkforsvar	dkmedier	wearenato	gensyn
Week 16	dkpol	dkmedier	dkforsvar	bidrag	enhedslisten
Week 17	dkpol	dkmedier	dkforsvar	forsvar	eupol
Week 18	dkpol	dkforsvar	dkmedier	enhedslisten	wearenato
Week 19	dkpol	dkforsvar	dkmedier	overmagt	forsvarsminister
Week 20	dkpol	dkmedier	dkforsvar	topmøde	stærkere
Week 21	dkpol	dkmedier	dkforsvar	wearenato	forbandedehyklere
Week 22	dkpol	dkmedier	dkforsvar	indsats	isil
Week 23	dkpol	konti	dkmedier	dkforsvar	twitter
Week 24	dkpol	dkmedier	letland	dkforsvar	eupol
Week 25	dkpol	dkmedier	dkforsvar	spørgelyst	truslerne
Week 26	dkpol	dkmedier	stemnej	dkforsvar	stem
Week 27	dkpol	dkmedier	dkforsvar	finland	sverige
Week 28	dkpol	stemnej	dkmedier	asger	dkforsvar
Week 29	dkpol	stemnej	dkmedier	eudk	dkforsvar

Table 7: Weekly top 5 words for topic 0 in the Danish dataset.

	Word 1	Word 2	Word 3	Word 4	Word 5
Week 0	stillingen	karantenetiden	fortbli	forbudatomvåpen	moetene
Week 1	frontex	attentatet	oppbevare	militærtribunal	utsultet
Week 2	i2	øvingsfelt	bygdnes	finnmarkskysten	norgemay
Week 3	kjendisstatus	nasjonalitetene	balkanhalvøya	propagandautspill	partipolitiker
Week 4	tøffinger	gryende	lyset	atomvåpenforbudet	innta
Week 5	realistiske	grei	attentater	konspisiljøene	higgins
Week 6	russlandhatere	rehabilitert	mellomstadier	vidtrekkende	russiskfiendlig
Week 7	elimineres	gasstilførselen	energisikkerhet	møtet	statsoverhoder
Week 8	aksepterer	forbrytelser	innlegget	evenes	utenriksnyheter
Week 9	hendte	vitende	offensiv	putinutrygghet	jasmå
Week 10	abonnement	glem	taliban	forfremmet	kontratrekkene
Week 11	paradistilstander	jenkins	ubeskyttet	bråke	debattensom
Week 12	he	norwegian	politiker	aksepterer	kvalifisert
Week 13	stråmenn	handler	muslim	talking	territoriell
Week 14	kypros	bothsideism	oppland	jo	lovet
Week 15	putin	våpen	nato	usa	russland
Week 16	sv	heller	oslo	putin	redaksjonskomitéen
Week 17	utvider	easy	talking	virkelighetsfjernt	østover
Week 18	millitæret	målrettede	zelenskyj	diktaturer	funker
Week 19	aksent	engelsken	korea	krigen	sikre
Week 20	håpe	sivile	mere	ga	nyhetene
Week 21	fortjente	soros	george	sittet	kriget
Week 22	serbere	albanske	heller	folkemord	su
Week 23	ensrettingen	krigssone	analytikere	ukrainia	borgerkrig
Week 24	fx	fredagsquiz	verdenspolitikk	apriletter	soge
Week 25	indisiene	fleste	støtter	terrorisme	peker
Week 26	sverige	tyrkia	finland	erdogan	skanske
Week 27	røyste	rødt	sur	knekke	drill
Week 28	kortere	rus	oppretting	iforpliktet	naboskap4

Table 8: Weekly top 5 words for topic 0 in the Norwegian dataset.