







## Article

# Sentence Embeddings and Semantic Entity Extraction for Identification of Topics of Short Fact-Checked Claims

Krzysztof Węcel , Marcin Sawiński , Włodzimierz Lewoniewski , Milena Stróżyna , Ewelina Księźniak   
and Witold Abramowicz \* 

Department of Information Systems, Poznań University of Economics and Business, 61-875 Poznań, Poland; krzysztof.wecel@ue.poznan.pl (K.W.); marcin.sawinski@ue.poznan.pl (M.S.); wlodzimierz.lewoniewski@ue.poznan.pl (W.L.); milena.strozyna@ue.poznan.pl (M.S.); ewelina.ksieznia@ue.poznan.pl (E.K.)

\* Correspondence: witold.abramowicz@ue.poznan.pl

**Abstract:** The objective of this research was to design a method to assign topics to claims debunked by fact-checking agencies. During the fact-checking process, access to more structured knowledge is necessary; therefore, we aim to describe topics with semantic vocabulary. Classification of topics should go beyond simple connotations like instance-class and rather reflect broader phenomena that are recognized by fact checkers. The assignment of semantic entities is also crucial for the automatic verification of facts using the underlying knowledge graphs. Our method is based on sentence embeddings, various clustering methods (HDBSCAN, UMAP, K-means), semantic entity matching, and terms importance assessment based on TF-IDF. We represent our topics in semantic space using Wikidata Q-ids, DBpedia, Wikipedia topics, YAGO, and other relevant ontologies. Such an approach based on semantic entities also supports hierarchical navigation within topics. For evaluation, we compare topic modeling results with claims already tagged by fact checkers. The work presented in this paper is useful for researchers and practitioners interested in semantic topic modeling of fake news narratives.

**Keywords:** sentence embeddings; semantic indexing; DBpedia; topic modeling; fake news; narratives; fact checking claims



**Citation:** Węcel, K.; Sawiński, M.; Lewoniewski, W.; Stróżyna, M.; Księźniak, E.; Abramowicz, W. Sentence Embeddings and Semantic Entity Extraction for Identification of Topics of Short Fact-Checked Claims. *Information* **2024**, *15*, 659. <https://doi.org/10.3390/info15100659>

Academic Editor: Kostas Vergidis

Received: 27 August 2024

Revised: 3 October 2024

Accepted: 8 October 2024

Published: 21 October 2024



**Copyright:** © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Today, the emergence, or even flood, of fake news on the Internet is being observed. It spreads rapidly through social media platforms, reaching a wide audience, and making it difficult to control its impact. This phenomenon undermines the public's trust in the media and creates a climate of uncertainty and confusion among Internet users. Additionally, fake news can manipulate public opinion and influence political discourse, posing a threat to democratic processes and societal harmony.

The identification and debunking of fake news by fact-checking agencies has emerged as a critical defense against the spread of misinformation on the web. This process is typically carried out by professionals associated with fact-checking organizations, such as PolitiFact, Snopes, and FactCheck.org. Their job involves several steps, starting with the identification and selection of a check-worthy claim. The claim is then verified by generating a fact-check rating based on available evidence, and finally a fact-checking report (called a debunk) is composed and shared with a wider audience.

Considering that this process is usually performed manually and the volume of information requiring verification is growing, there is a pressing need for effective and efficient IT solutions that can facilitate the work of fact checkers. This support is particularly crucial in two areas: (1) finding potential fake news to verify, as it involves monitoring numerous information sources, such as social media platforms, messaging services, microblogging platforms, and traditional media; and (2) identifying disinformation narratives (trending

or hot topics) among incoming information that should receive special attention from fact checkers.

The purpose of the research was to design a method to assign semantic topics to claims for their easier identification and tracking by fact-checking agencies. Considering that access to more structured knowledge is necessary during fact checking, it was necessary to study various semantic vocabularies, such as DBpedia, Wikipedia topics, YAGO, and other relevant ontologies. Furthermore, we required the method to group short claims by similarity, and here, sentence embeddings and various clustering approaches were deemed to be useful. Finally, the most important terms to describe topics had to be determined, so several methods based on classical TF-IDF as well as classification models for feature importance elicitation were employed.

The remainder of this paper is organized as follows. Section 2 presents the state-of-the-art in text representation, clustering techniques, and topic detection. Section 3 describes the research methodology, followed by the presentation of the method and the research results in Section 4. In Section 5, the findings are discussed. The paper concludes with suggestions for future work and discusses limitations.

## 2. Background

Hirlekar and Kumar [1] review and examine various methodologies and tools for fake news identification. While there are different approaches, they all share a common focus on feature extraction methods. The potential of advanced deep learning techniques for classification of fake news was more recently described in [2]. Here, we focus on a modern approach to text understanding, i.e., using embeddings for semantic similarity and classification.

An embedding refers to a learned representation of data that captures relevant features or characteristics of the data in a lower-dimensional space. In the NLP domain, it can represent a portion of text. It can be a word without context like in word2vec [3] or a context-dependent word piece as in BERT (Bidirectional Encoder Representations from Transformers) [4,5]. Word embeddings can find their application as an efficient classifier for fake news detection [6]. SBERT [7] is a modified BERT model that allows the execution of some specific NLP tasks at the sentence level.

These embeddings can further be used to group similar text segments, usually by means of cosine distance. They also find their use in topic modeling [8]. DBSCAN (Density-Based Spatial Clustering of Applications with Noise) [9] is a flexible algorithm for cluster identification. It exists in many varieties that address some issues of its canonical version [10]. A hierarchical approach is given in [11]. K-means implemented in the FAISS [12] technique is dedicated to being used with embeddings.

In order to speed up clustering, one can apply various space reduction techniques as a pre-processing step. Principal Component Analysis (PCA) [13] is one of the most prominent linear methods, whereas t-SNE (t-Distributed Stochastic Neighbor Embedding) [14] and UMAP (Uniform Manifold Approximation and Projection) [15] represent a non-linear class. In PCA, the reduction is achieved by identifying the principal components, which are orthogonal directions that capture the maximum variance in the data. t-SNE transforms objects from the high-dimensional space to the low-dimensional space by modeling pairwise similarities between data points in the input and output sets, emphasizing the preservation of local relationships. The UMAP approach is the generalization over t-SNE with the main difference in the assumptions about the space characteristics, which instead of Euclidean is a Riemannian manifold.

Classical approaches to topic modeling consist mainly of LDA (Latent Dirichlet Allocation) and NMF (Non-negative Matrix Factorization), whereas the more modern approaches include such models as Top2Vec [16] and BERTopic [17]. The last model was also used for fake news analysis (use case COVID-19) in [18]. LDA is a probabilistic topic modeling technique. It assumes that each document in a collection is a mixture of various topics and that each topic is a mixture of words. LDA aims to uncover these latent topics by analyzing

word co-occurrence patterns. By clustering documents based on their embeddings, Top2Vec identifies topic vectors that represent groups of similar documents. BERTopic is a topic modeling algorithm that utilizes the language model. It employs BERT's contextual word embeddings to create document embeddings and then applies clustering techniques to group similar documents into topics. However, it does not allow us to extract relevant phrases. BERTopic can handle large and diverse document collections, offering both efficient and accurate topic recognition. The comparison of the results generated by the models is given in [19,20].

Particularly challenging is the identification of topics for short texts. Our claims are usually contained within one sentence; therefore, we decided to compare our results to various topic modelings for short texts presented in [21] (we used the software available from <https://github.com/qiang2100/STTM> [21]). For example, the Dirichlet multinomial mixture (DMM) assumes that each text is sampled from only one latent topic. Some variations of the approach, e.g., GPUDMM and GPUPDMM, use another sampling process—the generalized Pólya urn (GPU)—hence the name. Biterm Topic Modeling (BTM) is an example of global word co-occurrences-based methods, where a biterm is formed from two words. In order to mitigate the sparsity of word co-occurrences, some methods merge short texts into long pseudo-documents, like Self-aggregation-based Topic Modeling (SATM) [22], Pseudo-document-based Topic Modeling (PTM) [23], and the Word Network Topic Model (WNTM) [24]. NMF factorizes a term-document matrix into two non-negative matrices, one representing the topics and the other representing the document-topic distribution. By iteratively updating these matrices, NMF identifies the underlying topics and their distribution within the documents.

Our approach differs from those presented above as we strive to combine embeddings with semantic entities. We propose to leverage hierarchical dependencies among semantic entities, mostly superclasses in DBpedia and Wikidata. The most challenging part is to identify the most important entities best describing the specific topic. For this task, a TF-IDF analysis of terms may be performed [25]. The TF-IDF weighting scheme was originally used in the information retrieval field to identify keywords relevant to the studied corpus [26]. It is similar to the task of topic recognition with the main difference being that in the latter case, the set of topics is given, whereas topic modeling assumes unbounded generation of keywords on the text analysis basis. There are numerous methods to achieve the goal of topic modeling. A method must provide the topic keywords that characterize properly the content of a document, and specifically, in the context of document clustering, it should allow the grouping of semantically similar texts. The task of topic modeling can be augmented by knowledge taken from a knowledge base. This effort was made using Wikidata [27] or DBpedia [28]. It is worth noting that the mentioned embedding approach may also be applied to knowledge bases. KEPLER is a unified model for Knowledge Embedding and Pre-trained Language Representation [29]. Their embeddings seem to be attractive, but they were trained purely on descriptions of entities. According to the authors, KEPLER maintains a strong language understanding ability and additionally incorporates factual knowledge. They show how KEPLER works as a Knowledge Embedding model and evaluate it on Wikidata5M, a large-scale knowledge graph constructed from Wikidata and Wikipedia. Several available pre-trained models were prepared using Wikidata5m [https://graphvite.io/docs/latest/pretrained\\_model.html](https://graphvite.io/docs/latest/pretrained_model.html) accessed on 31 July 2024.

Research similar to ours presented in this paper was also initiated by another team. The conclusion from [30] was that topic modeling can improve the accuracy and interpretability of fake news detection. However, their analysis was restricted to the classical LDA method, and ours considers more topic modeling methods along with a different way to represent terms, i.e., semantic instead of just words.

### 3. Method

In this section, we describe the main methodological justification behind our research. We delineate the process from claim collection to generation of terms that describes identified fake-related topics.

#### 3.1. Collection of Claims

Claims for the purpose of the experiment were collected with the APIs of the Google Fact Check Tool (<https://toolbox.google.com/factcheck/apis>). We have downloaded all claims since 2001 until March 2023. The number of retrieved claims is 39,409. We included only claims in English and further excluded claims that could not be verified based purely on text, i.e., not by looking at pictures or videos. An example of such an unverifiable claim is: “Video shows: «statement»”. After cleaning, our corpus contains 24,315 claims. The breakdown of claims by agencies is presented in Table 1.

**Table 1.** Number of claims in our dataset broken down by fact-checking agencies.

Fact Checker	Number of Claims
Snopes.com	7000
PolitiFact	6068
AFP Fact Check	4308
FactCheck.org	2245
USA Today	1964
The Washington Post	1216
POLYGRAPH.info	963
Newsweek	504
others	47
Total	24,315

#### 3.2. Semantic Embedding

Each claim in the collection contains two textual fields: `title` and `claim_text`. The usage of these fields depends on the fact checker. For example, Snopes.com puts questions in the title and answers in the claim text: “Is this California ‘ASN FLU’ License Plate Real?” and “A vanity license plate with a configuration of letters that spelled out ‘ASN FLU,’ pronounced Asian flu, was issued in California”. PolitiFact includes answer in the title to make it explicit, for example: “No, Bernie Sanders did not collaborate with Marxist regimes” and the claim text contains debunked statement “Say Bernie Sanders «collaborated with Marxist regimes in the Soviet Union, Nicaragua and Cuba»” We considered only `claim_text` as a carrier of meaning and calculated sentence embeddings using `all-mpnet-base-v2` model (<https://huggingface.co/sentence-transformers/all-mpnet-base-v2>). This model maps text fragments to a 768-dimensional dense vector space and is specifically trained for clustering and semantic search for English.

#### 3.3. Clustering of Claims

We considered two types of methods for clustering of claims: distance-based and density-based. Both of them can find their justification in claim analysis.

In distance-based methods, for example, K-means, a claim belongs to the cluster created by the nearest centroid. Thus, the Euclidean distance decides about the boundaries of the clusters. Such an approach is appropriate for keeping a consistent set of claims for easier description but may not be sufficient for analysis of evolving fake news narrations. Let us consider an example of the Olympic Games in Paris, being currently the most exploited topic for fake news production. It can contain a broad set of topics, from water pollution to religious incidents. We then need to account for slowly changing narration, and focusing on a single topic while contributing claims can have a dynamic focus. Such changes can be better modeled with density-based methods.

In our work, we have used specific implementations of the above clustering methods. In order to focus on the most important dimensions and speed up calculations in both clustering approaches, we reduced the number of dimensions. In the case of the density-based method, we first applied UMAP (<https://umap-learn.readthedocs.io/en/latest/clustering.html>). By studying the content of obtained clusters, we determined the best hyperparameters: number of neighbors—15, number of components—5, and metric—cosine. We then applied HDBSCAN with the minimum cluster size—25, the metric—Euclidean, and the cluster selection method—expectation of mass. Visualization in a two-dimensional space was made possible by reducing space using a two-component UMAP.

One of the issues in the case of DBSCAN is unassigned claims—there are 11,918 (49%) such claims in this experiment. At first, one can consider such distribution problematic, but it makes sense from the fact-checking domain point of view. We required that the cluster consists of at least 25 claims because we are interested in narrations, and single statements are usually not significant for fact checkers. In the case of K-means, such outliers would be combined with the nearest cluster, making it less homogeneous.

In the case of distance-based methods, we decided to use the K-means implemented in FAISS (<https://faiss.ai/>), a tool for efficient indexing of embeddings, semantic search, and clustering dense vectors. Before training the K-means model in FAISS, we first applied principal component analysis (PCA) requiring that at least 90% of variance be explained, thus reducing 768 dimensions to 219. The optimum number of clusters using the Silhouette score was calculated as 2, which did not make sense for our application. Finally, to make results comparable, we set  $k = 80$ , similarly to the UMAP + DBSCAN approach.

The final calculation of the clusters was computationally complex. The wall time necessary for the calculation was around an hour. Effectively, the total CPU time was 2 days and 17 h (it was calculated on a machine with 32 CPUs/64 threads).

We have characterized the clusters obtained with several measures that are typically used to assess the quality of the clustering (see Table 2). The Silhouette coefficient compares the mean distance between a sample and all other points in the same cluster vs. the nearest cluster; higher values are better [31]. The Silhouette coefficient has higher values for PCA + K-means than for UMAP + HDBSCAN, which means that K-means detects better defined clusters. It is typical that the Silhouette coefficient is generally higher for convex clusters than for density-based clusters. When we remove unassigned claims from the analysis, the coefficient for UMAP + HDBSCAN is better (see second column). Such a phenomenon is not observed for the Calinski–Harabasz index, which prefers PCA + K-means. According to the Davies–Bouldin index, the clusters obtained with UMAP + DBSCAN are better separated.

**Table 2.** Measures characterizing various clustering approaches.

Measure	UMAP + HDBSCAN	UMAP + HDBSCAN corr	PCA + K-Means
Silhouette coefficient	−0.037	0.040	0.023
Calinski–Harabasz index	46.0	54.0	92.3
Davies–Bouldin Index	3.38	3.04	4.19

We also assessed mutual information between clusters built with UMAP + HDBSCAN and FAISS K-means. The normalized mutual information (NMI) between the clusters is 0.45. It is generally higher for two clusterings with a larger number of clusters, e.g., like 80 in our case. Therefore, adjusted mutual information (AMI) is adjusted for a chance and is 0.42, where random partitions have an expected AMI around 0.0. Another measure, the adjusted rand index (ARI), computes the similarity between two clustering results by considering all pairs of claims. Here, ARI is 0.03, which means that the labeling looks like it is random. The situation looks much better when we again remove unclassified claims. The measures are as follows: NMI = 0.71, AMI = 0.69, and ARI = 0.37.

### 3.4. Semantic Entity Extraction

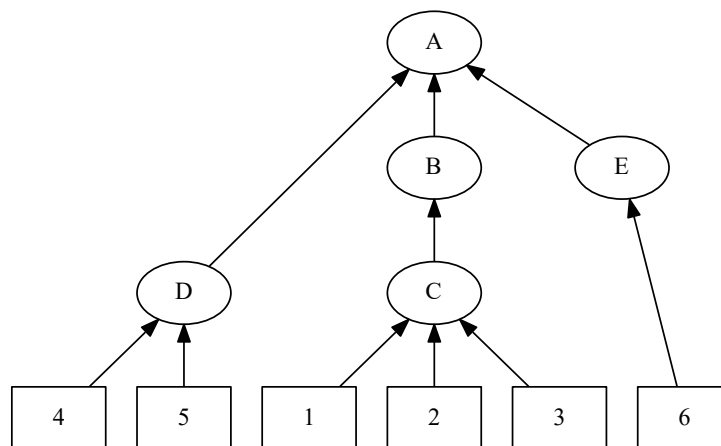
We used DBpedia Spotlight (<https://www.dbpedia-spotlight.org/>) to extract semantic entities (mentions) from claims. It is a system for automatically annotating text documents with DBpedia URIs and is described in detail in [32]. We have used their API for English language with default settings. Entities are detected with various confidences; therefore, we need to provide this value as argument to call; we have retrieved entities with varying confidences: 0.1, 0.2, 0.5, and 0.9. In this work, we use entities obtained with confidence 0.5. The sample entry, serialized in JSON, is presented below.

```
{ "id": 4274,
  "entity": [
    "http://dbpedia.org/resource/NATO",
    "http://dbpedia.org/resource/Madrid",
    "http://dbpedia.org/resource/Spain" ] }
```

The semantic entities are further processed—in ‘entity-to-term’ mapping where terms are then related with topics. We were interested in finding common topics for terms; therefore, we leveraged a taxonomy of semantic entities and assigned them to terms generalizing additional entities (e.g., superclasses). The following schemes for generalization were applied:

- `dbpedia(entities_only)`—only DBpedia entities as extracted from claims are used
- `dbpedia(rdf_type_all)`—DBpedia instances are mapped to classes by following `rdf:type`
- `dbpedia(rdf_type_no_yago)`—same as above but all classes from YAGO ontology are excluded
- `dbpedia(rdf_type_yago)`—only superclasses from YAGO are used
- `dbpedia(rdf_type_ontology)`—only classes from DBpedia ontology are used
- `wikidata(all)`—similarly to `rdf:type`, classes from Wikidata are used along with their extensions by following Wikidata’s P279 property for all superclasses
- `wikidata(no_equivalents)`—same as above but in the case of equivalent classes, only one class is considered, e.g., either <http://dbpedia.org/ontology/Person> or <http://www.wikidata.org/entity/Q5> (access date 31 July 2024).

The idea of deduplication is as follows. We only want to keep classes that introduce new information. This is in line with the general objective to look for semantic entities spanning multiple claims. Only classes that span multiple subbranches are interesting, i.e., their in-degree is greater than one. All other classes are just repeating information. For example, in Figure 1, we present instances denoted with numbers and classes denoted with letters. Class B is redundant because it will have exactly the same instances as class C.



**Figure 1.** Hierarchy of classes.

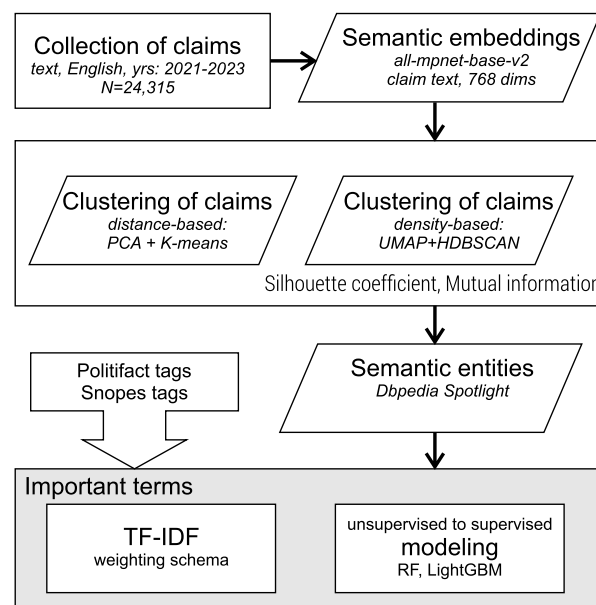
### 3.5. Identification of Important Terms

Having indexed all claims with terms obtained from semantic entities in one of the seven ways described above, we can now move on to the main process of this work. In the

next step, we group all claims into clusters that were established based on embeddings, using one of the described methods (UMAP + HDBSCAN, PCA + K-means). Experiments were carried out in variants: all 24,315 claims and separate calculations for PolitiFact and Snopes where tags of fact checkers were used as a golden standard. We obtained matrices where the rows were representing merged claims (equivalent to clusters), and the columns were terms. The goal was to determine the most important terms for the topics.

Simple measures based on term frequency are not sufficient; therefore, we decided to leverage TF-IDF weighting schema. Twenty terms with the highest weight were chosen from each cluster as their representatives. They were a foundation for topics that can be better identified by fact checkers. In addition to TF-IDF, we applied another approach based on ‘unsupervised to supervised’ modeling. For each cluster, which was detected using the unsupervised method, we build a classifier that should be able to distinguish a given cluster from all other clusters, i.e., one versus all. We then used the built-in methods of the classifiers to determine which features (here, terms) were the most important in the classification. Two classifiers were used: Random Forest (RF) and LightGBM (LGBM). We also tested XGBoost during the initial experiment but LightGBM was an order of magnitude faster and provided very similar results (i.e., 15 s vs. 2–3 min).

The block schema of the overall approach is presented in Figure 2.



**Figure 2.** Workflow of the system.

## 4. Experiment and Results

### 4.1. Evaluation

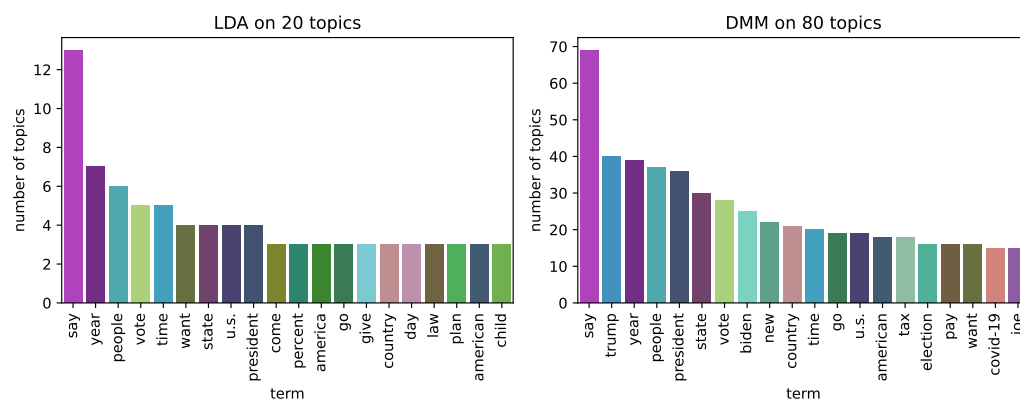
We compare the results of our method with various topic modeling methods, including those for short texts: LDA, DMM, BTM, WNTM, PTM, SATM, GPUDMM, and GPUPDMM. Three variants of the number of topics are considered: 20 (close to the number in HDBSCAN), 40, and 80 (to make it comparable with K-means). Evaluation is conducted for two fact checking agencies for which we could obtain tags for claims: PolitiFact and Snopes. As they usually use many tags, we also applied different methods to reduce the multi-label problem to a single label: (1) take all tags as a single tag, resulting in a big number of pseudo-classes [taga], (2) take just the first tag, assuming it to be the best described content [tagff], and (3) take the most popular tag among the mentioned tags, resulting in the smallest number of classes [tagpf]. Three aspects of topic modeling performance are considered: coherence, clustering, and classification. For coherence, we use point-wise mutual information (PMI): English Wikipedia articles are used as a reference for co-occurrence calculation. Clustering is assessed with NMI and purity; for both, higher values are better.

Normalized mutual information specifies the reduction in entropy (uncertainty) when the cluster labels are known. Purity effectively shows the percentage of the most frequent classes in respective clusters to all data points. Finally, the classification aspect is covered by determining the accuracy of the SVM model with fivefold cross-validation. The higher the accuracy, the more discriminative and representative the learned topics are.

In order to assess the alignment of terms and topics, we also calculated the most central claims. It works in a similar way to selecting sentences for summarization. More specifically, we calculated the average embedding of all claims in a cluster, the so-called centroid. Then, we retrieved five claims with the lowest distance from the centroid. For faster retrieval, we leveraged FAISS, which had been used before for semantic similarity.

#### 4.2. Uniqueness of Words in Topics

One of the first observations is that in the case of reference (baseline) topic modeling methods, words are very often repeated within topics. This means that they are not discriminative for topics and are of little help to fact checkers. Almost all methods identify the word ‘say’ as the most popular in most topics. For example, in the PolitiFact subset, it occurs in 13 topics out of 20 in the case of LDA, and in 69 topics out of 80 in the case of DMM, which was designed to work much better for short texts (see Figure 3).



**Figure 3.** Distribution of words among topics for PolitiFact subset modeled with LDA with 20 topics and DMM with 80 topics.

Our proposed methods do not yield such repetitions. For example, for terms extracted directly from claims in the PolitiFact subset, the most popular words, like election or budget, are repeated in only 5 out of 80 topics for K-means. For the same subset, DBSCAN returned another set of terms, but they described a smaller number of topics (25). Words generalized with ‘dbpedia-(entities\_only)’ are repeated more often, but this is caused by the smaller number of terms for indexing, i.e., only identified named entities can contribute, not all words. The discussed results are presented in Figure 4. The extended results are presented in figures in Appendix A.

Our idea was, nevertheless, the generalization of semantic entities, i.e., we intended to describe topics with the most specific generalizing classes. For example, instead of writing ‘Trump’ and ‘Biden’, we can write ‘Politician’. Such a generalization has a trade-off for the description of topics—terms are repeated more often. Considering generalization ‘dbpedia(rdf\_type\_ontology)’ where entities are replaced with their respective types, the term ‘dbo:Person’ was used to describe 28 out of 80 topics (see Figure 5). Such a behavior would be expected if topics were characterized by named people. Nevertheless, topics can still be very well characterized as several terms and their intersections need to be considered, e.g., topic ‘Person and China’ is different from ‘Person and Election’.



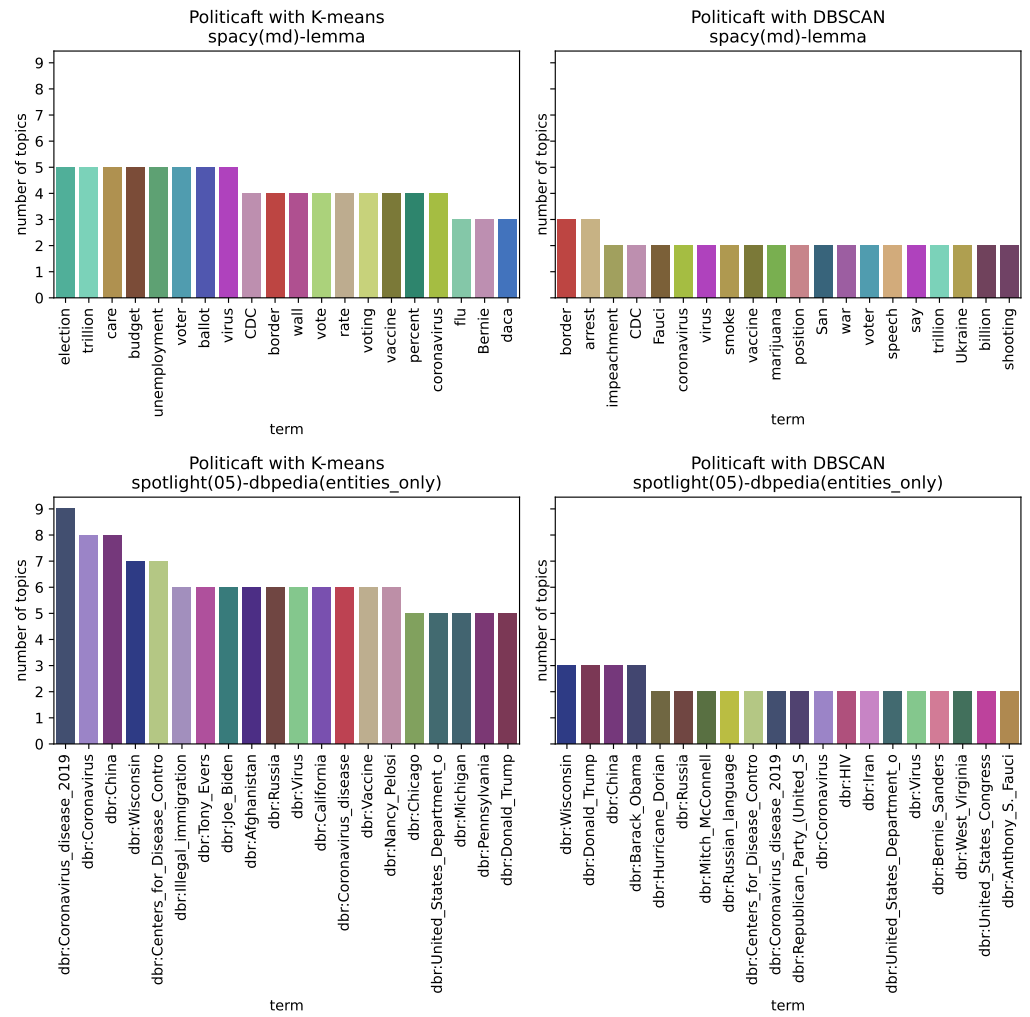


Figure 4. Distribution of terms among topics for custom methods.

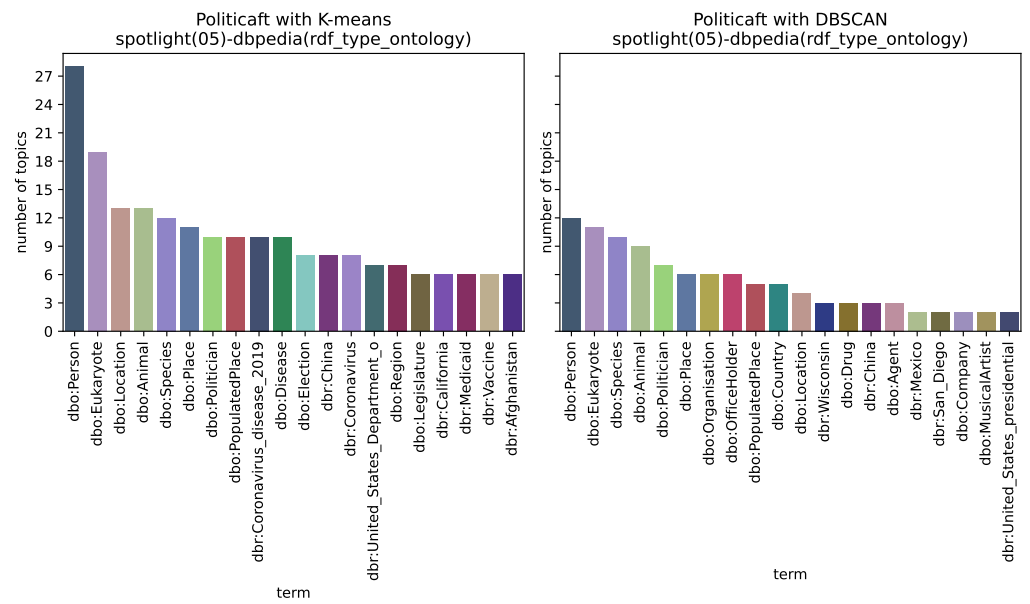


Figure 5. Distribution of ontology terms among topics for custom methods.

### 4.3. Classification Evaluation

Likewise, for classification, measured by accuracy, the number of topics did not have a big impact. The most important thing was how we chose the tags with which we wanted to compare our topics—the tag set. PolitiFact was tagged with 165 unique tags, and only 144 tags were used as the first tag (“tagff” in the figure). When we consider all tags assigned by fact checkers to a claim, we obtain 2799 highly fragmented tag combinations (“taga”), which produce results with lower accuracy. The accuracy of our proposed method ‘faiss(80)’ is better by a small margin. Regarding accuracy, clustering based on K-means outperformed DBSCAN in all cases (see Figure 6).

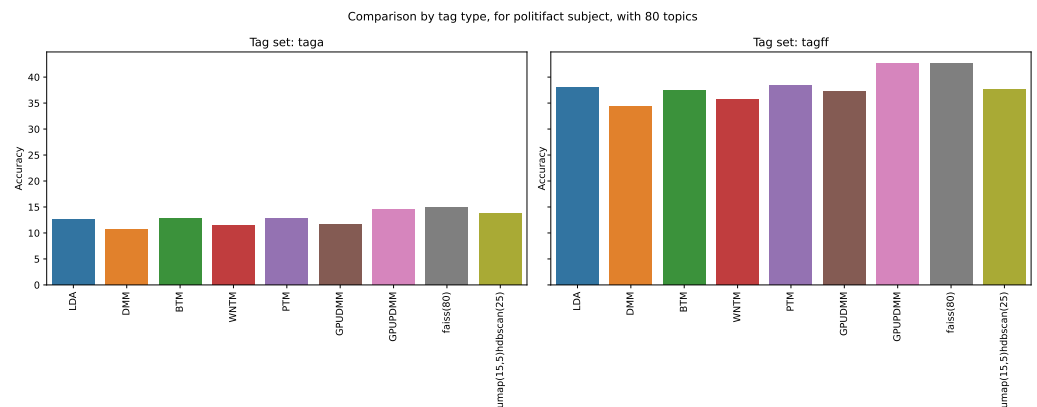


Figure 6. Accuracy of various topic modeling methods, for PolitiFact with 80 topics.

Our custom method’s accuracy is better by a small margin. Regarding accuracy, clustering based on K-means outperformed DBSCAN in all cases (see Figure 7).

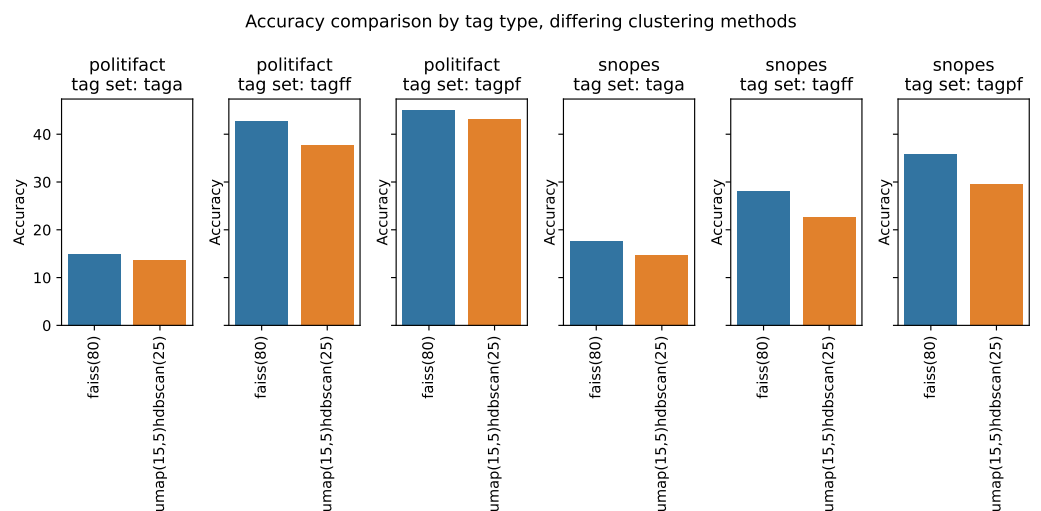


Figure 7. Accuracy of custom methods, for various tag assignment approaches.

### 4.4. Coherence Evaluation

The coherence of our two datasets PolitiFact and Snopes, was comparable between methods. SATM performed the worst, achieving 1.08 and 1.01, while WNTM was the best, scoring 1.21 and 1.16, respectively (see Figure 8). Similar results were achieved for 40 and 80 topics.

In the case of our methods, the coherence score was more diverse and depended on the generalization method (see Figure 9). Plain terms did not perform well—although word representation was not repeated so often, the coherence was low. The best results were achieved for dbpedia(entities\_only), regardless of the clustering method.

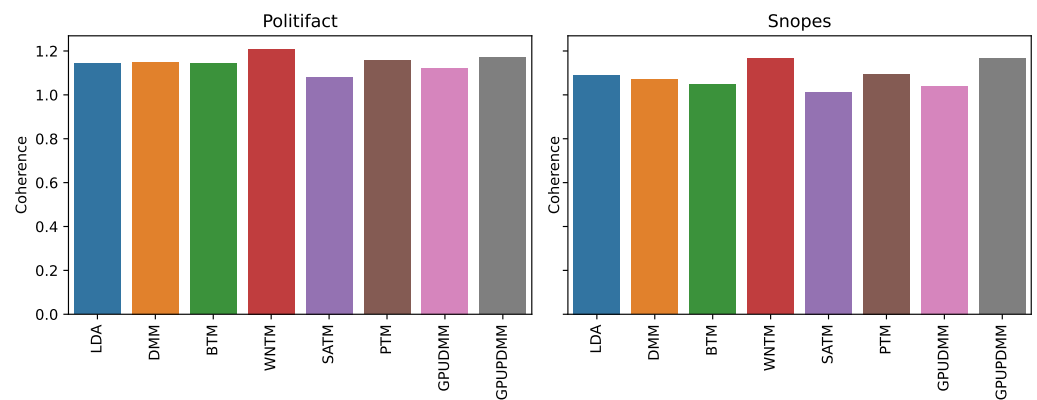


Figure 8. Coherence of 20 topics for various topic modeling methods.

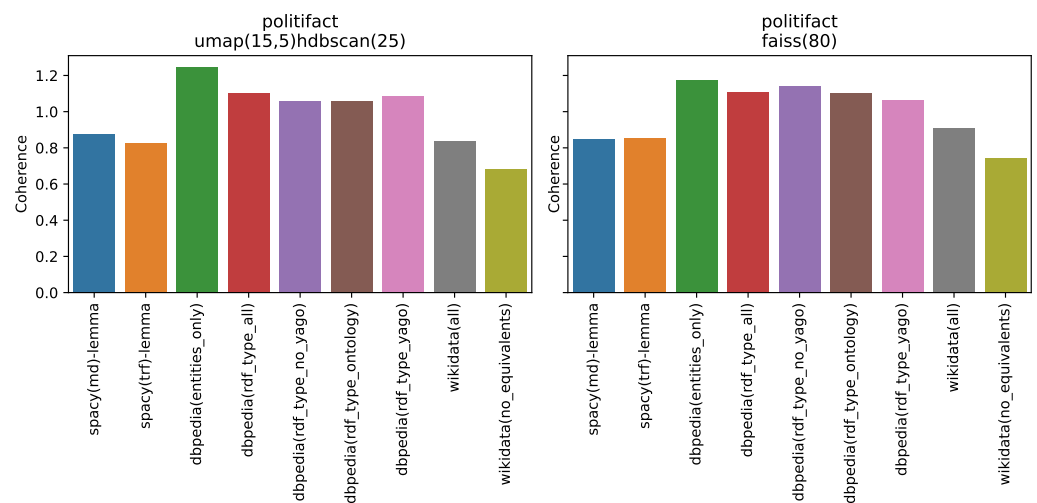


Figure 9. Coherence of topics produced by our methods.

Concluding, we have conducted coherence evaluation to make sure that the proposed method based on semantic entities is not worse than state-of-the-art methods based on terms. We obtained comparable results, regardless of the number of described topics and clustering method. As we also needed to advise on the best representation method, results are in favor for semantic entities with DBpedia URIS, without additional reasoning on higher-level concepts.

### 5. Discussion and Future Work

In this paper, we presented our approach to assigning semantic topics to claims published by fact checkers. The ultimate goal is to support fake news detection by providing contextual information, or even further, by preparing federated queries to respective knowledge graphs.

Based on the claim itself, it is hard to assess which aspect should be used to formulate the main topic and how to classify it for further fact checking. In the studied examples, very often, we encountered geographical entities (e.g., countries) accompanied by a subject (e.g., war, vaccination). However, when we properly aggregate many claims, such a distinction will be easier as we will know where the critical mass goes. Our method was designed to identify generalizations that would promote the hottest topics among the debunked claims. Moreover, by using density-based clustering, it becomes even more apparent which topics were indeed dense with possible anchors for fake narrations.

In our evaluation, we showed that our method, which combines sentence embedding and semantic enrichment, performs better than known topic modeling methods for short texts with regard to the identification of relevant terms. We also demonstrated that it does not perform worse with regard to commonly used metrics for topic evaluation.

During our research, we identified the need for an experimentation tool that would allow one to quickly verify our partial results. Thus, the added value is provided by the application for the visualization of clusters and displaying the characteristics of clusters. We used two clustering methods (DBSCAN and K-means), seven entities to terms mappings, and three weighting schemes for the assessment of feature importance (TF-IDF, Random Forest, LightGBM).

The contributions of this paper are as follows. We proposed to use semantic entities to describe topics, so that semantic relations between entities can be generalized and descriptions simplified. The paper also contributes a method to assign semantic entities to short-text claims, by the mean of combining many such texts based on similarity calculated from sentence embeddings. We verified two embedding clustering approaches and concluded that there are no significant differences in the results, although density-based methods may be better suited for fake news narration tracking. We conducted a qualitative analysis of topics created using various methods, both classical and for short texts. We also carried out a quantitative analysis of coherence between topics, and other commonly used metrics for topic evaluation. For generalization of semantic entities, the most explainable results were provided by DBpedia ontology. We suggest avoiding Wikidata projects, as they were mostly ambiguous and too general. For the ranking, TF-IDF and RF provided similar results. We advise against using LGBM as this method; although fast, it tends to promote top-level classes, losing focus on details.

For future work, we plan to apply a new clustering method (KwikBucks), propose another method for generalization based on shorter paths in a knowledge graph, and evaluate weighting schemes using Random Forest and LightGBM. It would be interesting to focus on hierarchical clustering. Semantic entities are organized hierarchically, so this would be a natural extension as well.

## 6. Limitations

The limitations of our approach can best be discussed using the steps mentioned in the methodology. The first concern is the representation of claims in vector space, namely how embeddings are calculated. In order to make semantic search and clustering possible, the embedding model needs to be trained in a specific way. There are not many such models, and their accuracy could still be improved. Moreover, the models we used were trained for English and our texts are also in English. Extension to other popular languages would be necessary. In the next experiment, we will apply the E5-family of embeddings.

The next issue to discuss is clustering. Based on our experience and observations of the work of fact checkers, we carried out analysis for a various number of topics—20, 40, 80—expecting various required specificities of classification. The bigger number of clusters is not expected by fact checkers. We calculated clustering quality measures and this indicated that our method does not seem to be sensitive with regard to the number of topics. Both DBSCAN and K-means provided comparable results, but there are other methods on the horizon, for example “KwikBucks: Correlation Clustering with Cheap-Weak and Expensive-Strong Signals”, presented at ICLR 2023 (<https://openreview.net/pdf?id=p0JSSa1AuV>).

For semantic entity extraction, we used DBpedia Spotlight. Although we used a relatively low confidence level (0.5), not all entities were extracted. Sometimes, it was caused by the wording. There are also observed problems with generalization of extracted entities. YAGO ontology seems to be too detailed and provides too many possible variants (multiple inheritance). We also studied Wikidata with regard to ‘instance of’ (P279) and ‘subclass of’ (P31) properties. These are not standardized in any way and we discovered

many inconsistencies. Promising initial results were achieved when occupation (P106) was considered. More focused studies on coming from entities to their generalizations in Wikidata would also improve results.

Finally, in this paper, we only showed the results of one weighting scheme, namely TF-IDF. There are, however, other options, such as using the importance of variables from classification models. We have already carried out experiments for Random Forest, XGBoost, and LightGBM, and they will be presented in another paper.

**Author Contributions:** Conceptualization, K.W., M.S. (Marcin Sawiński), and W.L.; methodology, K.W.; software, M.S. (Marcin Sawiński), W.L. and K.W.; validation, M.S. (Milena Stróżyna) and E.K.; data curation, W.L. and M.S. (Milena Stróżyna); writing—original draft preparation, K.W.; writing—review and editing, K.W., M.S. (Marcin Sawiński), E.K., M.S. (Milena Stróżyna), W.L. and W.A.; visualization, K.W.; supervision, W.A.; project administration, W.A. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Center for Research and Development (NCBR, Poland), grant number INFOSTRATEG-I/0035/2021-00 OpenFact—Tools for verifying the credibility of information sources and detecting false information using artificial intelligence methods.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflicts of interest.

### Appendix A. Additional Figures

Figure A1 presents the first part of the full set of term frequencies for two datasets, whereas Figure A2 presents the second part.

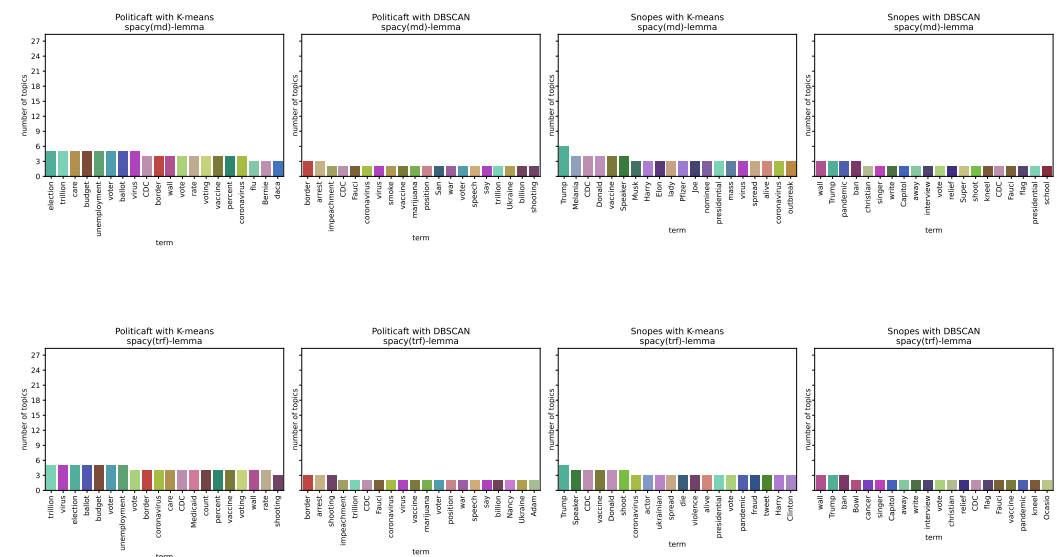


Figure A1. Cont.

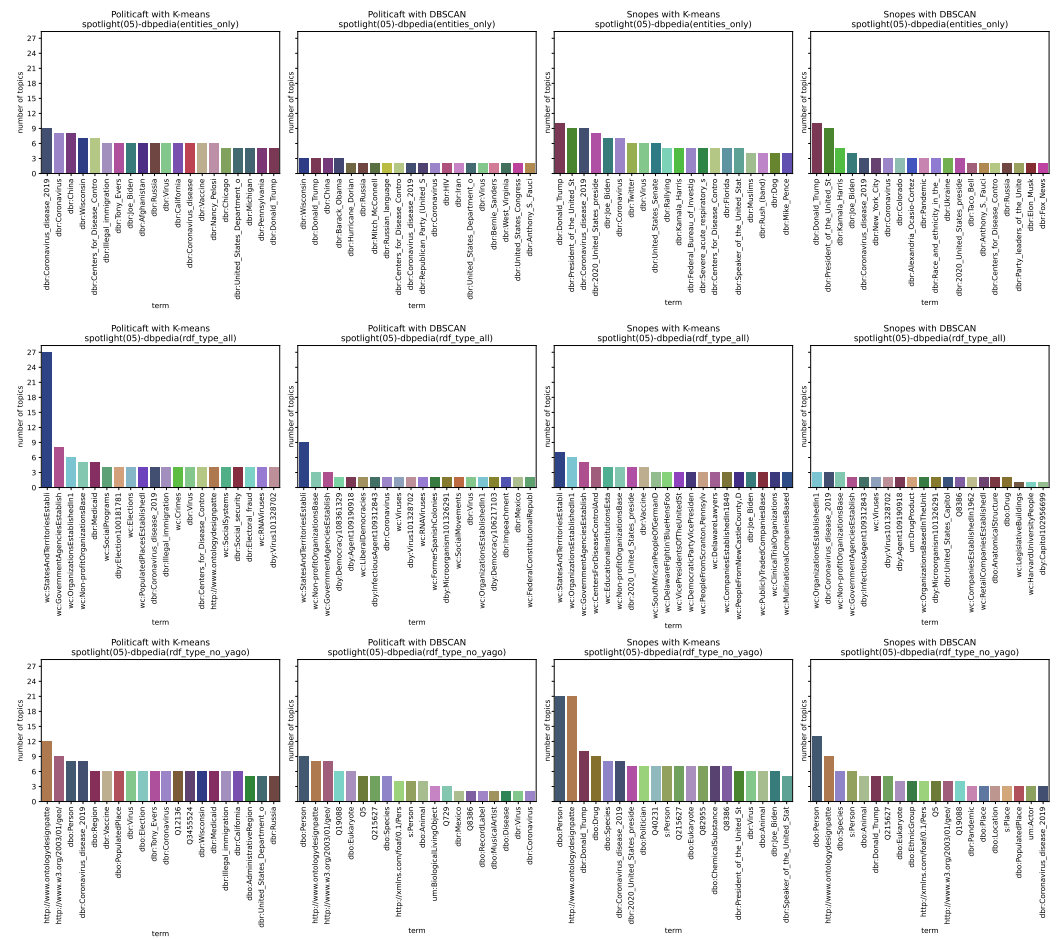


Figure A1. Full set of term frequencies for two datasets, with two clustering methods and various generalization schemes, part 1. The same terms are encoded with the same color.

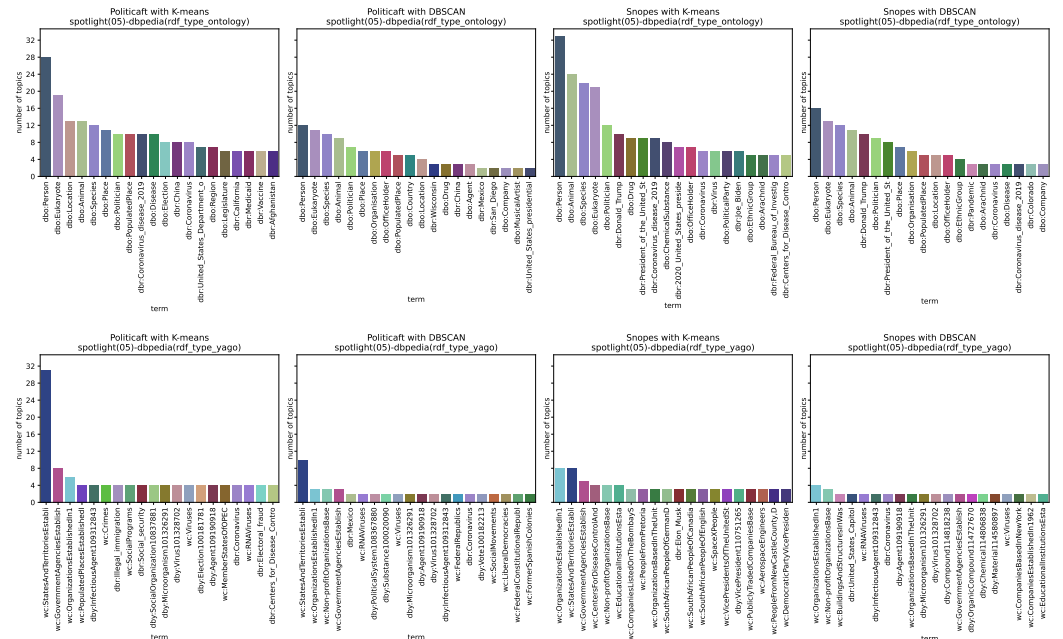
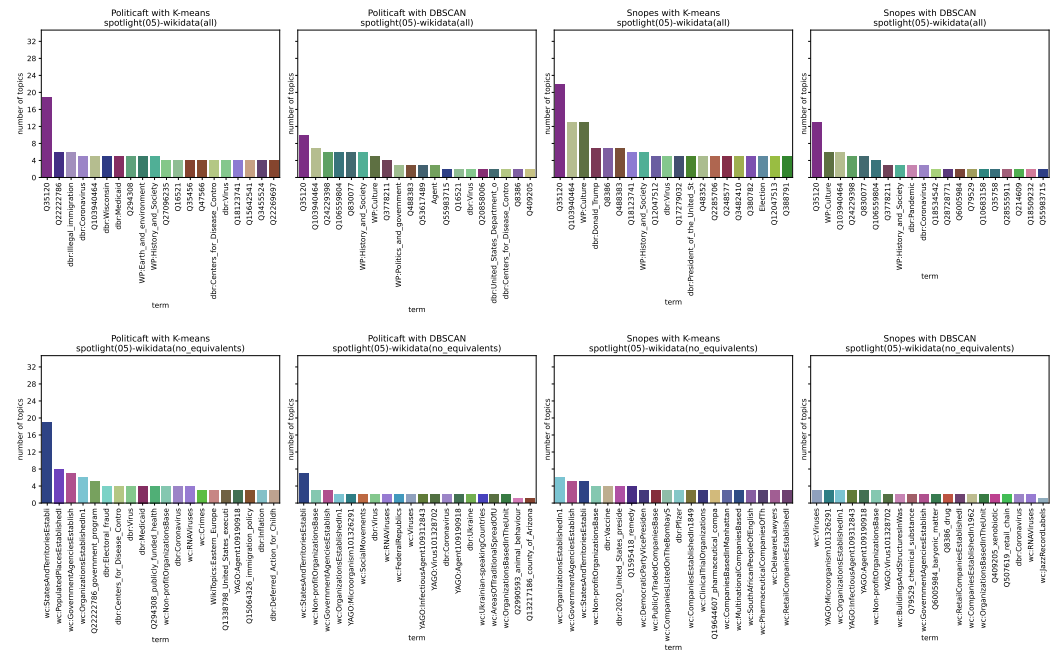


Figure A2. Cont.



**Figure A2.** Full set of term frequencies for two datasets, with two clustering methods and various generalization schemes, part 2. The same terms are encoded with the same color.

### Appendix B. Computing Times

Calculation times for datasets with claims were negligible. Below, we provide calculation times for a much bigger dataset with 209,811 news items from the HuffPost dataset <https://huggingface.co/datasets/khalidalt/HuffPost>. Topic models were calculated both for new headlines (shorter texts) and news descriptions (longer texts; hence, also longer calculation times). Table A1 presents the calculation times for 20 topics.

**Table A1.** Calculation times for different topic modeling methods on 20 topics. Times are provided in CPU hours.

Method	Head	Description	Ratio
LDA	0.08	0.1	1.25
DMM	0.10	0.13	1.33
BTM	0.20	0.67	3.33
WNTM	0.72	3.73	5.21
SATM	1.87	3.80	2.04
PTM	1.03	1.47	1.42
GPUDMM	1.05	1.47	1.40
GPUPDMM	39.73	55.95	1.41
LFLDA	41.73	53.27	1.28
LFDMM	26.52	inf	n/a

### References

- Hirlekar, V.V.; Kumar, A. Natural Language Processing based Online Fake News Detection Challenges—A Detailed Review. In Proceedings of the 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020; pp. 748–754. [\[CrossRef\]](#)
- Padalko, H.; Chomko, V.; Chumachenko, D. A novel approach to fake news classification using LSTM-based deep learning models. *Front. Big Data* **2024**, *6*, 1320800. [\[CrossRef\]](#) [\[PubMed\]](#)
- Goldberg, Y.; Levy, O. word2vec Explained: Deriving Mikolov et al.’s negative-sampling word-embedding method. *arXiv* **2014**, arXiv:1402.3722.
- Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
- Jwa, H.; Oh, D.; Park, K.; Kang, J.M.; Lim, H. exbake: Automatic fake news detection model based on bidirectional encoder representations from transformers (bert). *Appl. Sci.* **2019**, *9*, 4062. [\[CrossRef\]](#)

6. Al-Tarawneh, M.A.B.; Al-ir, O.; Al-Maaitah, K.S.; Kanj, H.; Aly, W.H.F. Enhancing Fake News Detection with Word Embedding: A Machine Learning and Deep Learning Approach. *Computers* **2024**, *13*, 239. [CrossRef]
7. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084.
8. Miranda-Belmonte, H.U.; Muñoz-Sánchez, V.; Corona, F. Word embeddings for topic modeling: An application to the estimation of the economic policy uncertainty index. *Expert Syst. Appl.* **2023**, *211*, 118499. [CrossRef]
9. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, KDD'96, Portland, OR, USA, 2–4 August 1996; pp. 226–231.
10. Khan, K.; Rehman, S.U.; Aziz, K.; Fong, S.; Sarasvady, S. DBSCAN: Past, present and future. In Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Chennai, India, 17–19 February 2014; pp. 232–238. [CrossRef]
11. McInnes, L.; Healy, J.; Astels, S. hdbscan: Hierarchical density based clustering. *J. Open Source Softw.* **2017**, *2*, 205. [CrossRef]
12. Jégou, H.; Douze, M.; Johnson, J.; Hosseini, L.; Deng, C. Faiss: Similarity Search and Clustering of Dense Vectors Library. Astrophysics Source Code Library, Record ascl:2210.024. 2022. Available online: <https://ui.adsabs.harvard.edu/abs/2022ascl.soft10024J/abstract> (accessed on 31 July 2024).
13. Drikvandi, R.; Lawal, O. Sparse principal component analysis for natural language processing. *Ann. Data Sci.* **2023**, *10*, 25–41. [CrossRef]
14. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
15. McInnes, L.; Healy, J.; Melville, J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv* **2018**, arXiv:1802.03426.
16. Angelov, D. Top2Vec: Distributed Representations of Topics. *arXiv* **2020**, arXiv:2008.09470.
17. Grootendorst, M. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv* **2022**, arXiv:2203.05794.
18. Schäfer, K.; Choi, J.E.; Vogel, I.; Steinebach, M. Unveiling the Potential of BERTopic for Multilingual Fake News Analysis—Use Case: COVID-19. *arXiv* **2024**, arXiv:2407.08417v1.
19. Chen, W.; Rabhi, F.; Liao, W.; Al-Qudah, I. Leveraging State-of-the-Art Topic Modeling for News Impact Analysis on Financial Markets: A Comparative Study. *Electronics* **2023**, *12*, 2605. [CrossRef]
20. Egger, R.; Yu, J. A topic modeling comparison between LDA, NMF, top2vec, and BERTopic to demystify twitter posts. *Front. Sociol.* **2022**, *7*, 886498. [CrossRef]
21. Jipeng, Q.; Zhenyu, Q.; Yun, L.; Yunhao, Y.; Xindong, W. Short Text Topic Modeling Techniques, Applications, and Performance: A Survey. *arXiv* **2019**, arXiv:1904.07695.
22. Quan, X.; Kit, C.; Ge, Y.; Pan, S. Short and sparse text topic modeling via self-aggregation. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015), Buenos Aires, Argentina, 25–31 July 2015; pp. 2270–2276.
23. Zuo, Y.; Wu, J.; Zhang, H.; Lin, H.; Xu, K.; Xiong, H. Topic modeling of short texts: A pseudo-document view. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2016), San Francisco, CA, USA, 13–17 August 2016; pp. 2105–2114.
24. Zuo, Y.; Zhao, J.; Xu, K. Word network topic model: A simple but general solution for short and imbalanced texts. *Knowl. Inf. Syst.* **2016**, *48*, 379–398. [CrossRef]
25. Zhou, Z.; Qin, J.; Xiang, X.; Tan, Y.; Liu, Q.; Xiong, N.N. News Text Topic Clustering Optimized Method Based on TF-IDF Algorithm on Spark. *Comput. Mater. Contin.* **2020**, *62*, 217–231. [CrossRef]
26. Zhang, W.; Yoshida, T.; Tang, X. A comparative study of TF\* IDF, LSI and multi-words for text classification. *Expert Syst. Appl.* **2011**, *38*, 2758–2765. [CrossRef]
27. Lim, K.H.; Karunasekera, S.; Harwood, A.; Falzon, L. Spatial-based topic modelling using wikidata knowledge base. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 4786–4788.
28. Zarrinkalam, F.; Fani, H.; Bagheri, E.; Kahani, M.; Du, W. Semantics-enabled user interest detection from twitter. In Proceedings of the 2015 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT), Singapore, 6–9 December 2015; Volume 1, pp. 469–476.
29. Wang, X.; Gao, T.; Zhu, Z.; Zhang, Z.; Liu, Z.; Li, J.; Tang, J. KEPLER: A Unified Model for Knowledge Embedding and Pre-trained Language Representation. *arXiv* **2019**, arXiv:1911.06136. [CrossRef]
30. Hosseini, M.; Javadian Sabet, A.; He, S.; Aguiar, D. Interpretable fake news detection with topic and deep variational models. *Online Soc. Networks Media* **2023**, *36*, 100249. [CrossRef]
31. Rousseeuw, P.J. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *J. Comput. Appl. Math.* **1987**, *20*, 53–65. [CrossRef]
32. Daiber, J.; Jakob, M.; Hokamp, C.; Mendes, P.N. Improving Efficiency and Accuracy in Multilingual Entity Extraction. In Proceedings of the 9th International Conference on Semantic Systems (I-Semantics), Graz, Austria, 4–6 September 2013.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.