Coded Term Discovery for Online Hate Speech Detection

Dhanush Kikkisetti
American University
vk4372a@american.edu

Raza Mustafa
Loyola University New Orleans
rulmust@loyno.edu

Wendy Melillo
American University
melillo@american.edu

Roberto Corizzo
American University
rcorizzo@american.edu

Zois Boukouvalas
American University
boukouva@american.edu

Jeff Gill
American University
jgill@american.edu

Nathalie Japkowicz
American University
japkowic@american.edu

***Abstract*—Online hate speech proliferation has created a difficult problem for social media platforms. A particular challenge relates to the use of coded language by groups interested in both creating a sense of belonging for its users and evading detection. Coded language evolves quickly and its use varies over time. This paper proposes a methodology for detecting emerging coded hate-laden terminology. The methodology is tested in the context of online antisemitic discourse. The approach considers posts scraped from social media platforms, often used by extremist users. The posts are scraped using seed expressions related to previously known discourse of hatred towards Jews. The method begins by identifying the expressions most representative of each post and calculating their frequency in the whole corpus. It filters out grammatically incoherent expressions as well as previously encountered ones so as to focus on emergent well-formed terminology. This is followed by an assessment of semantic similarity to known antisemitic terminology using a fine-tuned large language model, and subsequent filtering out of the expressions that are too distant from known expressions of hatred. Emergent antisemitic expressions containing terms clearly relating to Jewish topics are then removed to return only coded expressions of hatred.**

***Index Terms*—hate speech, coded terminology, antisemitism**

I. Introduction

Online hate speech detection[1] is a complex problem for social media platforms. A particular challenge, not much discussed in the literature, relates to the use of coded language. The following post illustrates the issue in the context of antisemitic hate speech:

> "Nope Globalist want us intertwined and run by the elites, Globalist don't lay tariffs on their friends you stupid fu****". [posted on Dec. 31, 2022, on the Disqus platform]

According to the American Jewish Committee (AJC) Translate Hate Glossary[2], a *globalist*, in its unbiased definition, is "a person who advocates the interpretation or planning of economic and foreign policy in relation to events and developments throughout the world". According to this definition, the term

Fig. 1. Non antisemitic use of the term "globalist"

is rather flattering. Indeed, that is the way it is intended in the Hyatt hotel's welcoming message to its club members seen in Figure 1[3]. In that commercial context a globalist refers to someone "who gets it!" and should feel good about it! The term does not, in any way, refer to Jews.

Yet, the AJC Translate Hate Glossary argues that the term has an antisemitic connotation when it "is used to promote the antisemitic conspiracy that Jewish people do not have allegiance to their countries of origin, like the United States, but to some worldwide order—like a global economy or international political system—that will enhance their control over the world's banks, governments, and media". In the above post, it is clear that the antisemitic connotation is implied. From this post, we surmise that

1) The globalists (a.k.a., the Jews) are distinct from "us", presumably, the good American citizens;
2) They control "our" fate to be run by the elites (a subset of these Jews)[4];
3) They help each other by not imposing the same tariffs on each other as those they impose on "us".

The above post, thus has a double meaning. To a recipient who is unaware of its antisemitic connotation, some category

Published in IEEE DSAA'2024. **We gratefully acknowledge the financial support of American University's Signature Research Initiative Project**.

[1]Warning: Some of the paper's content may be disturbing to the reader.

[2]https://www.ajc.org/translatehate/globalist

[3]Photo by one of the authors on 11/3/23 at a Hyatt Texas property.

[4]"Elite" appears in the AJC Glossary in the context of "Cosmpolitan Elite".

of people, the *globalists*, do not seem to behave very nicely. Yet, to an informed audience, it is a very pointed post that reiterates old Nazi and Soviet antisemitic propaganda[5] and propagates it further. Furthermore, on social media platforms, it does so without setting off any serious alerts since, except for the "stupid fu****" mention, which could raise a flag, no offensive terms are used.

Awareness of coded terminology similar to *globalists* and *cosmopolitan elite* which carry both a "regular" and an antisemitic connotation depending on the context in which they are used is necessary for accurate hate speech detection as will be demonstrated in Section III. However, coded terminology evolves quickly as shown by the increase in terms published by the AJC's Translate Hate glossary between February 2021and today[6]. Yet, discovering such terms manually is time-consuming and yields incomplete results (see [12]). This paper is, thus, concerned with the automatic discovery of "coded" terminology. It proposes an automated monitoring tool to assist human monitors by suggesting emerging, potentially coded, antisemitic terminology, along with the posts that use that terminology.

Though the topic of hate speech is, unfortunately, quite vast, this study focuses on antisemitism. The choice of a particular category of hate speech comes from the assumption, confirmed in Section III, that we can perform a more thorough analysis of the problem by remaining focused. Antisemitism was selected because of the reported increase in antisemitic incidents in the months preceding the beginning of this study, in 2022, and its continuing increase. The lessons learned from this particular study will apply to other categories of hatred including hatred against Black, Muslim, Asian, and LGPTQ+ populations amongst others.

The main contribution of this paper is a methodology for the problem of extracting *emerging coded hate-laden terminology* from extremist social posts, along with a practical pipeline to demonstrate its effectiveness. The idea is to harvest terminology used in similar contexts as known antisemitic terminology and propose it as potential emerging antisemitic coded terminology to human monitors, along with the context in which that terminology occurs. We propose four different versions of our pipeline and validate them using a quantitative approach. The most successful version is also evaluated qualitatively.

The remainder of the paper is structured as follows: Section II presents background and related work. Section III shows the results of a preliminary study that demonstrates that the two main assumptions at the basis of our work—1) generalized hate speech detection underperforms when considered on specific kinds of hatred and 2) knowledge of coded terminology can significantly boost the performance of hate speech detection—actually hold. In Section IV, we discuss how the data used in this research was collected and pre-processed. Next, the methodology and pipeline for extracting coded terminology is introduced in detail in Section V. This is followed by a presentation and discussion of the results in Section VI. Finally, Section VII concludes the paper and discusses future work.

II. Background and related work

With the advent of the internet and social media, technology has increased the speed at which language evolves. Propaganda in the form of hate speech now travels the world at such a fast pace that it is beyond human capacity to keep up with. Harmful words take on new meanings in both direct and coded ways, inciting hatred in the minds of those only too willing to believe them as they reinforce and justify preexisting prejudices.

A. Machine Learning for Hate Speech Detection

In recent times, there has been a notable rise in hate crimes across the United States.[7] While establishing a clear relationship between hate crimes and online content is not straightforward, a report by the US Department of Justice points to the simultaneous purchases of Facebook ads containing dividing content and hate crime. These two reports[8] thus suggest that hate speech should not be considered harmless, and coming up with methods to curb it is an important goal.

Previous work aims to detect hate speech from social media using various Machine Learning (ML) methods as documented by a number of surveys written in the last six years [3], [7], [15], [17]. One of the most recent surveys shows that while up to 2016, fewer than 10 papers were published on the topic each year, since then, there has been a huge increase in interest in the topic with over 150 papers published in 2020, the last year for which their survey had complete information [7]. Hate speech detection has been attempted using a wide variety of techniques and applied to many different problems. Founta et al. [4], for example, used Recurrent Neural Networks (RNN) to classify racism and sexism. Serrà et al. [19] showed that character level based Long Short-Term Memory networks (LSTMs) for abusive language detection could be useful. Similarly, Convolutional Neural Networks (CNNs) have also been shown to be successful in hate speech detection and classification [5]. More recently, transformer-based large language models have been used for these tasks like in the work of [20] who propose different fine-tuned and non-fined-tuned variations of pre-trained models such as BERT, RoBERTa, ALBERT, etc. on offensive language detection. Most of these studies, however, consider hate speech as a whole and, typically, do not distinguish the community towards which it is directed. We assume that this generalized approach is too broad and decide, instead, to use a divide-and-conquer approach by focusing on particular communities separately. Our first attempt focused on the Jewish community and the problem of antisemitic speech in social media. Section III shows that, indeed, the state-of-the-art generalized hate speech detection models do not perform that well on antisemitic

[5]c.f. "Globalist" and "Cosmopolitan Elite" in the AJC Glossary.

[6]See, 2021: https://www.ajc.org/sites/default/files/pdf/2021-02/AJC_Translate-Hate-Glossary-2021.pdf vs. Current: https://www.ajc.org/translatehateglossary

[7]https://bjs.ojp.gov/library/publications/hate-crime-recorded-law-enforcement-2010-2019

[8](1) https://bit.ly/2xeeF5h; (2) https://www.ojp.gov/pdffiles1/nij/grants/304532.pdf

speech detection, thus, justifying our focused approach. We are in the process of applying our research to islamophobia detection and will, then, extend it to anti-Asian speech.

B. Antisemitism in Social Media and its Detection

Antisemitism specifically targets Jewish individuals or the Jewish community [18]. In [21], authors use the outcomes of two surveys from the EU and the ADL to assess how the level of antisemitism relates to the perception of antisemitism by the Jewish community in eight different EU countries. Furthermore, a recent survey finds that 20% of American Jewish adults have experienced an act of antisemitism, such as an attack either online or on social media.[9] In another study, the authors address the challenges of quantifying and measuring online antisemitism. It raises the question of whether the number of antisemitic messages is increasing proportionally to other content or if the share of antisemitic content is rising. Additionally, the paper aims to determine the extent of online Jew-hatred beyond well-known websites, forums, and closed social media groups [8].[10]

A few studies have attempted to combat online antisemitism in a way similar to the way in which generalized hate speech has been countered in the works discussed in the previous section. In [1], for example, the authors prepared a data set that includes both social posts and associated images, when available. They labeled the entries as antisemitic or not, and if antisemitic, indicated the kind of antisemitism: political, economic, religious or racial. They used a bimodal deep learning approach for classifying the data into these categories. [2] considers a subset of the text-only part of this dataset in an attempt to classify antisemitic posts using a less computationally-intensive approach. Focusing on the class imbalance problem in the data while taking advantage of OpenAI's GPT technology, they compared GPT-based resampling techniques against other traditional kinds. Very recently, [9] proposed a new data set for antisemitism detection in social media posts that uses a strict annotating process. The data set is so recent, however, that it has not yet been used for classification or the results obtained on such efforts have not yet been published. There are other projects that consider the detection of online antisemitism using AI approaches as well. In particular, the project entitled "Decoding Antisemitism"[11] calls itself an "AI-driven Study on Hate Speech and Imagery Online", and already produced five published reports on the subject. The project specifically aims at linking national or international events reported in the traditional media to antisemitic online social media discussions.

C. Alternatives to automated hate speech detection

In [14], the authors question whether the way in which hate speech detection has been handled by the machine learning community is the way forward, or whether hate speech detection is a lot more complex than previously assumed by the researchers who labeled data sets and applied classifiers on them. Furthermore, the authors note that some hateful content may occur without the use of well-known slurs and that on top of it all, the nature of hate speech is constantly evolving.

Our work takes these observations into consideration and focuses on the particular problem of fast-evolving subtle hateful content that may elude automated monitoring systems and, in certain circumstances, human monitors. The purpose of our work is to, both, enhance the performance of automated hate speech detection approaches and help related social science studies deal with the avalanche of data they must process, to produce relevant, up-to-date results. This paper, specifically, aims to provide an approach for the detection of emerging antisemitic, coded (as well as non-coded), terminology used on extremist social media platforms.

TABLE I
COMPARISON OF STATE-OF-THE-ART MODELS FOR HATE SPEECH DETECTION ON A TASK OF ANTISEMITIC HATE SPEECH DETECTION.

Model	Precision	Recall	F1-Measure	Accuracy
Bert-base-uncased	**0.82**	**0.83**	**0.83**	**0.83**
Bertweet	0.74	0.71	0.72	0.73
HateBert	0.75	0.71	0.71	0.73
bertweet-hate-speech	0.71	0.69	0.69	0.70

III. PRELIMINARIES

This section presents a preliminary study whose purpose is to justify and motivate the aim of our main study. We do so through a series of simple experiments. We use the data set described in the next section and divide it into a training and a testing set. We begin by comparing the performance of four different contextual sentence embedding extraction models on the task of antisemitism detection. Two of the models are state-of-the-art specialized hate speech models (HateBert and bertweet-hate-speech), one is specialized for tweets (Bertweet), while the last one is a generalized English language model (bert-base-uncased). In each case, after the sentence embedding extraction was performed, the classification layer was trained on the training set, and the resulting classifiers were applied to the testing set. The cross-entropy loss function was used to compute the loss and the Adam optimizer [10] was used to update the weights over four epochs. For this experiment, the training set contained 527 posts distributed as 380 antisemitic and 147 non-antisemitic posts and the testing set contained 132 posts with 95 antisemitic posts and 37 non antisemitic ones. Table I shows the performance of the four models. Interestingly, non-specialized bert-base-uncased performs better on this task than the models that were specifically designed for and pre-trained on tweets or on hate-speech detection tasks. This suggests that our idea of treating different kinds of hatred separately is justified. Indeed, our results show that general hate speech detection pre-trained models are not particularly apt at classifying antisemitic versus non-antisemitic content. This is a rather surprising result which

[9]https://bit.ly/41FV6ei

[10]These studies preceded 10/7/23 when the situation worsened drastically. They don't take into consideration, for example, the recent Spring 2024 campus protests against the war in Gaza that have, in certain cases, led to antisemitic rhetoric.

[11]https://decoding-antisemitism.eu/

TABLE II
FINE-TUNED BERT MODELS AT THE SENTENCE LEVEL, CODED-TERM LEVEL, AND A COMBINED SYSTEM THAT EMPLOYS BOTH APPROACHES

Method	Precision	Recall	F1-Measure	Accuracy
Sentence Level	0.9	0.83	0.86	0.81
Coded Term Level	0.73	0.97	0.86	0.73
Combined Method	**0.92**	**1**	**0.96**	**0.94**

may, perhaps, be explained by the fact that specialized hate-speech models promote particularly offensive terms at the expense of more subtle expressions of hatred.

In our next experiment, we assess the usefulness of knowing relevant coded terminology in the particular category of hatred under consideration. This is done in the context of fine-tuned bert-base-uncased (the best model in our previous experiment) and the combination of two models. In particular, we created two fine-tuned versions of bert-base-uncased. The first version, *Sentence Level*, used the standard sentence-level fine-tuning approach described in Section V-B1. The second version, *Coded Term Level*, used a method focused on coded terminology. It began by localizing the coded terms present in each post using the list of coded antisemitic terminology introduced in section IV and then used the pre-truncate approach (i.e., approach 1 in Phase 2 of the method described in Section V-B2) to form embeddings. All the coded term embeddings present in a particular post were concatenated to form a single embedding of that particular sentence. After forming the embeddings using either the Sentence level or Coded Term level method, we added a classifier layer and trained the models. A third approach, *Combined Method*, consisted of combining both versions using the "OR" logical operator as follows:

- If version 1 returns true; then return true
- If version 1 returns false; then if version 2 returns true, then return true; Otherwise, return false

The results in Table II show that the first method fine-tuned at the sentence level performs overall better than the second method fine-tuned at the coded term level. It is interesting to see, however, that they are complementary in that the first method obtains a high precision and lower recall whereas the second method obtains a relatively low precision but very high recall. The difference in recall makes sense given that the sentence level method is expected to miss the very subtle expressions of antisemitism which use seemingly neutral (but coded) terms. On the other hand, the coded-term level method catches these posts but probably misses the more obvious ones. As a result, the combined method obtains excellent results, as shown in Table II, as it combines the pros of each method. This result is important as it motivates our claim that staying "on top" of hate speech coded terminology is essential for capturing hate speech detection in a holistic manner.

Having justified the importance of focusing on a single kind of hatred rather than the general problem of hate speech detection and having demonstrated the importance of coded terminology, we now turn our attention to the discovery of emergent antisemitic coded terminology.

IV. DATA PREPARATION

This study is part of a large multi-disciplinary project sponsored by our institution which, simultaneously, collects and analyzes the use of coded language to express antisemitic sentiment in lightly moderated social media platforms typically preferred by individuals with extremist tendencies and studies the migration of this language from these extremist platforms to the general population. The overall project includes a lexical analysis team, a population analysis team, and a software design team which collaborate closely and work in parallel.

A. Dataset

The project is constantly evolving, though for this study, we considered the first delivery of the data curated by the lexical analysis team in June 2023. The lexical analysis team's objective concerning their study was to analyze the usage of known antisemitic terms in extremist social media and search for new coded or non-coded terminology using human analysts. The objective of the software design team was to conduct a similar search for antisemitic terminology using AI techniques. The lexical analysis team's point of departure was the Translate Hate Glossary published by the American Jewish Congress (AJC), an organization founded over a century ago to defend Jewish interests in America and advocate for equal rights for all Americans.[12]

1) Data Scraping and Labeling: To build the corpus, the lexical analysis team analyzed antisemitic social media posts from various extremist social media platforms including Discuss, Telegram, Minds, and GETTR. It used antisemitic expressions obtained from the previously mentioned American Jewish Committee (AJC) Translate Hate Glossary as well as the Southern Poverty Law Center (SPLC) to collect social media posts. This collection effort was facilitated by Pyrra[13], a private software company that allows its users to scrape posts from alt-social media platforms according to a list of seed terms. The data team considered the 46 seed expressions available from the AJC Glossary at the time (from the version of the Glossary released in February 2021) as well as the term "Cultural Marxism", discussed in a SPLC article[14] and chose 16 of them to make the process tractable. It analyzed the 659 retrieved posts related to these seed expressions to determine whether the post was antisemitic or not and to discover potentially new coded or non-coded antisemitic terminology. Expert lexical analysts were trained to identify antisemitic content as follows: seven historic antisemitic tropes were identified and terms related to each of these tropes were listed using the AJC Glossary. When analyzing a post, if the post contained one of the terms associated with a trope, its context

[12]The AJC currently advocates for freedom of speech, church-state separation, education, women's rights, LGBTQ+ rights, and the rights of Jews across the world.

[13]https://www.pyrratech.com/

[14]https://www.splcenter.org/fighting-hate/intelligence-report/2003/cultural-marxism-catching

was analyzed to see if it matched the context in which the AJC considers the use of the term as antisemitic. If so, the post was labeled as antisemitic. If the post did not contain any of the known terms but seemed antisemitic, the researchers were instructed to follow a six-step process to determine whether the post contained an emerging antisemitic term. If so, the post was labeled as antisemitic and the new term was listed as a potentially emergent antisemitic term. Each post was assigned to three coders who were instructed to follow four rules of conduct in case of a disagreement to see if they could reach a consensus. When no consensus could be reached, the post was flagged as unresolved. The 16 terms used in the subset were selected based on their potential to reveal posts that had emerging new antisemitic terms in them. The list of seed words used is: *Cabal, Cosmopolitan Elite, Cultural Marxism, Deicide, The Goyim Know, Holocough, Jewish Capitalist, Jewish Communist, Jew Down, Jewish Lobby, New World Order, Not the Real Jews, Rothschild, Soros, Zionist,* and *Zionist Occupied Government*. The details of the procedure is available in the lexical analysis team's coding statement that can be found in our GitHub repository along with the data set, code, and other documentation.[15]

2) Preprocessing: Text preprocessing is a critical step in Natural Language Processing (NLP). It involves transforming raw text data into a format that can be easily analyzed by machine learning algorithms. The preprocessing steps usually used involve several techniques, such as tokenization, stop word removal, stemming, and lemmatization [13]. During the first phase of cleaning the corpus, we removed the `urls` and lower-cased all the posts to normalize them. This initial procedure was followed by stop words removal. Then we lemmatized the text to get a single root form for each word prior to passing it on to the coded antisemitic terms extraction process, which will be discussed in the next section. Bigrams and trigrams were formed by running two- and three- word windows through all the posts.[16] It was important to filter out badly-formed expressions obtained through that approach. In particular, we decided to include bigrams and trigrams that only contain nouns, proper nouns, adjectives, and verbs, since others were judged less relevant to our quest. Since the emphasis of this study is on the novel proposed extraction process discussed next, we did not experiment with the various pre-processing techniques suggested in the literature on hate speech for social media posts [6]. It was left for future work.

V. Coded Antisemitic Terms Extraction Approach

As previously mentioned, based on the importance of coded terminology for hate speech detection and the fact that hate speech detection should be conducted separately for different kinds of hatred, we designed an approach to extract *emerging coded antisemitic terms*. In order to carry out this goal, we designed a method for operationalizing each term of that expression. That operationalization and the linking of its resulting components into a functional system constitute the main contribution of this work. The purpose of this section is to discuss the process. To begin with, we consider each word in the *emerging coded antisemitic terms* expression and give it the specific meaning shown below.

- **Terms:** the extracted expressions are limited to *grammatically consistent* bigrams and trigrams; they have to be *relevant* enough to the documents in which they appear and appear *frequently* enough in the corpus.
- **Antisemitic:** the candidate expressions have to be *semantically related* to antisemitic discourse.
- **Coded:** antisemitic expressions that contain terms relating to obvious Jewish concepts are removed.
- **Emerging:** already known coded antisemitic expressions are removed in order to concentrate on new terminology.

These operations are divided into two phases. In Phase 1, we address the extraction of frequent, well-formed, emergent, and coded terminology without worrying about its semantic relation to antisemitism. In Phase 2, we address semantics using large language models. The resulting pipeline that illustrates our proposed methodology is shown in Figure 2. In that figure, Phase 1 is represented by the "Important Terms Extraction" component while Phase 2 is represented by the combination of the Embeddings Generation, Similarity Scoring, and Antisemitic Terminology Extraction. Both phases of the pipeline are implemented using two approaches: Approach 1 and Approach 2. We subsequently test all four combinations, yielding Methods 1-1, 1-2, 2-1, and 2-2.

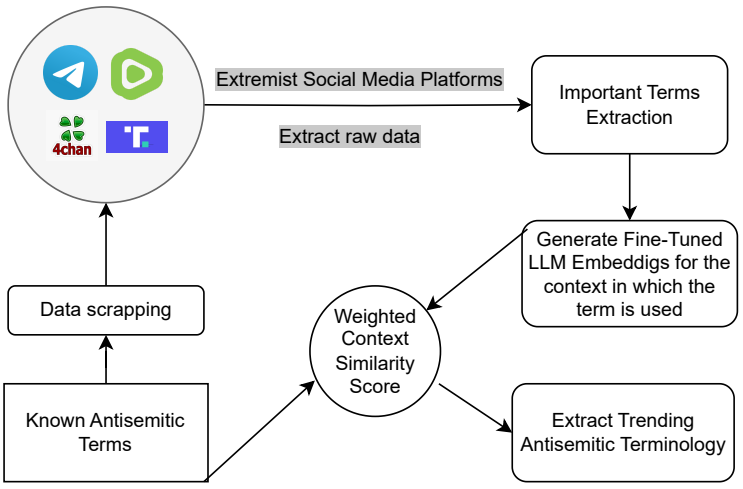

Fig. 2. Emergent Coded Antisemitic Terminology Extraction Pipeline

A. Phase 1: Emerging Coded Trending Terms Extraction

For the first part of Phase 1, the extraction of trending terms, we explore the use of off-the-shelf NLP tools for our first approach and then propose a second approach that combines tf-idf and frequency. Once the trending terms are extracted, we propose a strategy to remove non-emerging and non-coded

[15]https://github.com/dhanushk66/DSAA

[16]We also considered unigrams but were not able to filter them effectively using our current methodology. Their treatment was left for future work.

terms from the list of extracted terms. This strategy is applied to both Approaches 1 and 2.

1) Approach 1: Trending Terms Extraction using Concordance and Collocation tools: In this first attempt at trending terms extraction, we use traditional NLP techniques to extract bi-grams and tri-grams using concordance and collocation algorithms from the NLTK Toolkit [11]. Concordance is a technique that provides a comprehensive view of how a given term appears in a corpus. Using this approach, we use the 16 seed terms from Section IV-A1 for analyzing patterns and gaining insights into language usage. For each occurrence of a seed term, this approach provides the surrounding words context. We use default settings for the extraction of context. Next, using collocation, we find the most frequent bi-grams and tri-grams in the collected contexts. Collocation is a technique that finds a meaningful combination of words from a corpus that are semantically coherent. Different statistical measures can be used to detect collocations including frequency, pointwise mutual information (PMI), and log-likelihood ratio (LLR) among others. We use frequency, here since that is the measure also used in the advanced approach In the future, we plan to experiment with other statistical measures for both approaches. The standard approach yielded 126 trending terms.

2) Approach 2: Trending Terms Extraction using TF-IDF and Frequency: Our proposed second approach is presented in Algorithm 1 which uses TF-IDF feature-weighting [16] and frequency to extract trending terms. In a nutshell, this was done by selecting all the terms that obtained a TF-IDF value greater than a self-set threshold, listing these terms in decreasing order of frequency, and selecting the top 200 terms from the list.[17] When the same term appeared in several documents, the highest TF-IDF value it received was retained. Algorithm 1 shows the approach that was followed in detail. We explain its main steps below.

The algorithm begins by initializing the `Trending_terms` list which is the list that will return the 200 bigrams and trigrams (terms) that received the highest combination of TF-IDF and Frequency scores. Next, the TF-IDF values obtained for each unique term and each document are calculated and placed in the matrix $\mathbf{W}$ of size `d x v` where `d` represents the number of documents whereas, `v` is the number of terms. We then take each term in matrix $\mathbf{W}$ and find the highest score across all the rows (documents) of the matrix and store it in D_s. To remove the less relevant terms, we compute the average of all the values in D_s and use this value, δ, as a threshold. This allows us to consider only the terms with TF-IDF values greater than the average value of all the scores in D_s. If so, we save the terms' values and their frequencies in D_f. Finally, we sort the terms in D_f in descending order of their frequency values and select the top 200 terms, storing them in `Trending_terms`.

3) Removal Strategy: Redundant, Non-Emergent and Non-Coded terms Removal: Once the list of most trending terms has been extracted using either Approach 1 or Approach 2, three categories of terms are removed from it using the following automated procedures. First, as we consider expressions that are both bigrams and tri-grams, there is a possibility of encountering bigrams within trigrams. Such redundant bigrams are removed from the list of expressions. Next, we remove the terms that have occurred earlier. For now, this corresponds to the original list of 16 seed words used to retrieve the posts. In the future, this list will grow as we intend to use the system continuously, using newly discovered terms of interest as new seed terms. Lastly, the terms that are considered non-coded are removed. These correspond to terms that contain words that obviously pertain to Jewish themes. The list of words currently used includes jew, jewish, kike, and zionist, but it could be expanded. Expressions that include these words either as stand-alone words or embedded within other words are removed. All the removal procedures run automatically without any human intervention. After the removal phase is applied, we are left with 52 and 94 trending terms for Approach 1 and Approach 2 trending term extraction solutions, respectively.

[17]We assume that at least 200 terms had a TF-IDF value larger than the self-set threshold.

Algorithm 1 Trending terms extraction

Initialize Trending_terms ▷ Stores top *200* trending terms
Initialize D_s ▷ Stores terms' highest TF-IDF scores (s)
Initialize D_f ▷ Stores terms' values and frequencies
Set T ▷ Stores all the vocabulary terms (value).
Set F ▷ Stores the frequency of each term in the corpus.
Calculate the TF-IDF scores for each term in each document and store them in matrix $\mathbf{W} \in \mathbb{R}^{d \times v}$, where d denotes the number of documents and v denotes the vocabulary size.
for each term t in $\mathbf{T}$ **do**
  for each row in $\mathbf{W}$ **do** ▷ Each row is a document
    $D_s[t] \leftarrow max(D_s[t], W[row, t])$ ▷ Finds t's highest TF-IDF score, s, across all documents
  end for
end for
$\delta \leftarrow Average(D_s)$ ▷ δ is the average of all s's
i = 1
for each t in D_s **do**
  if $D_s[t] \geq \delta$ **then**
    $D_f[i] \leftarrow (T[t], F[t])$ ▷ If t's highest TF-IDF score is larger than threshold δ, store t's value and frequency in D_f
    i = i + 1
  end if
end for
Sort D_f in descending order of frequency
for $i = 1, 2, \ldots, 200$ **do**
  $Trending\_terms \leftarrow D_f[i][T]$ ▷ Store the most frequent terms in Trending_terms (drop the frequencies)
end for

B. Phase 2: Embeddings and Comparisons

Though the bigrams and trigrams extracted in the previous section are known to be trending, their semantics are unknown and, in particular, there is no information as to whether or not these terms are antisemitic. To find out which of these trending expressions are antisemitic, we compare the context in which they are used to the context in which the known antisemitic expressions are used. If a trending term appears in contexts similar to those in which seed expressions occur, it will be deemed antisemitic. Otherwise, it will be discarded as non-antisemitic. To compute embeddings for the trending and seed terms, we begin by training a pretrained BERT model. Since BERT was not specifically trained on instances of hate speech or antisemitism, we train it with additional data collected using the same seed expressions as before (see details below). This pre-trained version of BERT is then used to generate contextual embeddings for both the trending terms discovered in the last section and the seed terms used to extract posts. These embeddings will consequently be used to determine semantic similarity between new and seed terms. We present the details of BERT's pre-training followed by Embedding Approaches 1 and 2 that we implemented.

1) Training the pre-trained BERT model: The generalized BERT model does not possess domain-specific vocabulary, thus it is not capable of handling coded hate speech such as antisemitism. Indeed, when such out-of-vocabulary terms occur, they get broken down into smaller tokens for which embeddings are generated. These are treated as rare tokens, yielding unsatisfactory results. To avoid this issue, we fine-tune the BERT model using an additional 56K posts extracted using the same seed words as before on Pyrra, but excluding the 659 posts used in our study. We, thus, extend BERT's vocabulary from 30k to 55k tokens, and fine-tune it using the Masked Language Modeling (MLM) approach. MLM is a pre-training approach that masks a few tokens. The model is subsequently trained to predict the masked tokens from the words that surround them.[18]

2) Comparing Trending Terms to Seed Terms: To differentiate between antisemitic and non-antisemitic terms during Phase 2, we compare the trending terms' embeddings to the seed terms' embeddings using Cosine Similarity. We generate two types of embeddings following i) Approach 1, the pre-truncate embedding method and ii) Approach 2, the post-truncate embedding method. In pre-truncate embedding, we truncate the post containing the term to be embedded *prior to* embedding it. In post-truncate embedding, we embed the entire post containing the term of interest, and truncate the resulting embedding *afterwards*.[19] In order to ensure a representative context, we remove seed words that did not succeed in extracting a large enough number of posts using Pyrra. We set the threshold to 5 posts in this particular study. Two seed terms were excluded due to this policy—"Cosmopolitan Elite" and "Jew down"—leaving us with only 14 representative seed terms for the remainder of the study.

[18]https://huggingface.co/learn/nlp-course/chapter7/3

[19]Since we cannot embed posts exceeding 512 tokens, we turned large posts into multiple ones.

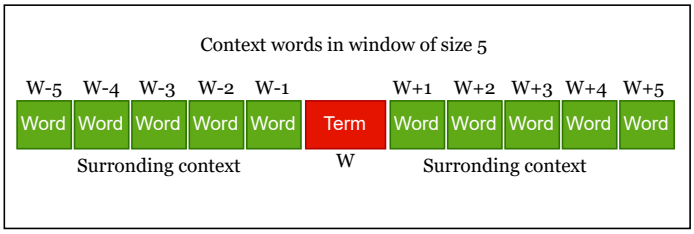

Fig. 3. Pre-truncate embedding approach for a window of size 5.

a) Approach 1: Pre-truncate embeddings: In this approach, we consider context windows of 5 to 14 words, where the size of the windows refers to the twin windows located before and after the term being embedded, respectively. We show an example of windows of size 5 in Figure 3. Since the same term may be found in more than one post, we concatenate all the embeddings extracted from fine-tuned BERT using the same window size and take their average. Embedding, here, refers to the `pooled layer` obtained from the 12 layers of the BERT architecture. We follow the same procedure for all the trending terms we extracted and the 14 seed words from Section IV-A1 that were retained as sufficiently representative.

Next, we determine whether the trending terms have an antisemitic connotation using Algorithm 2, described in general terms as follows. After some initializations, the "similarity to antisemitism" value for trending term tt, is computed as follows: tt's embedding is compared to each of the 14 representative seed terms (the st's)'s embeddings using Cosine Similarity. The 14 resulting measurements are then averaged and assigned to $S[tt]$. The process is repeated for each trending term and the median of all the $S[tt]$'s, γ, is calculated. γ is then used as our threshold for potential antisemitism as follows: if $S[tt]$ for trending term tt is greater than γ, tt will be given the partial label "potentially antisemitic" ($TT\_PL\_w[tt]$ = 1). Otherwise, it will be given the partial label "probably not antisemitic" ($TT\_PL\_w[tt]$ = 0). (We used the median as it offered more flexibility than the mean.) Algorithm 2 is repeated 10 times, once for each window size w considered. This yields 10 partial labels $TT\_PL\_w[tt]$, $w = 1 \ldots 10$ for each term tt, and the final labeling for tt is "antisemitic" if m out of the 10 partial labels are "potentially antisemitic". It is "not antisemitic", otherwise. The optimal value of m was 7 for the pre-truncate case.

b) Approach 2: Post-truncate embeddings: In this approach, we begin by embedding each complete post using fine-tuned BERT. The approach is illustrated in Figure 4 for an 18-word post. This yields an 18 x 12 x 768 tensor representing the total number of words in the post, the total number of encoding layers, and their dimension. This embedding can be thought of as a word embeddings lookup table that pro-

Algorithm 2 Comparing semantic similarity–window size w

$Embeddings\_tt \leftarrow \{et\_1, et\_2 \ldots, et\_n\}$ ▷ n pre- or post- truncate trending terms embeddings at window size w
$Embeddings\_st \leftarrow \{es\_1, es\_2 \ldots, es\_14\}$ ▷ 14 pre- or post- truncate seed words embeddings at window size w
Initialize TT_PL_w. TT_PL_w will store the n trending terms & predicted antisemitic label for window size w.
Initialize S. S will store the average semantic score for each trending term at window size w.
for each *tt* in **Embeddings_tt do**
    for each *st* in **Embeddings_st do**
        $tt\_scores[tt] \leftarrow Sim(et\_tt, es\_st)$ ▷ Cosine Sim
    end for
    $S[tt] \leftarrow Average(tt\_scores[tt])$ ▷ Average all the 14 semantic scores between tt and all the st's
end for
$\gamma \leftarrow Median(S)$ ▷ γ is the median of all the scores
for each *tt* in **S do**
    if $S[tt] > \gamma$ **then** ▷ check if score greater than γ
        $TT\_PL\_w[tt] \leftarrow 1$ ▷ if score greater than γ
    else
        $TT\_PL\_w[tt] \leftarrow 0$ ▷ if score less than γ
    end if
end for

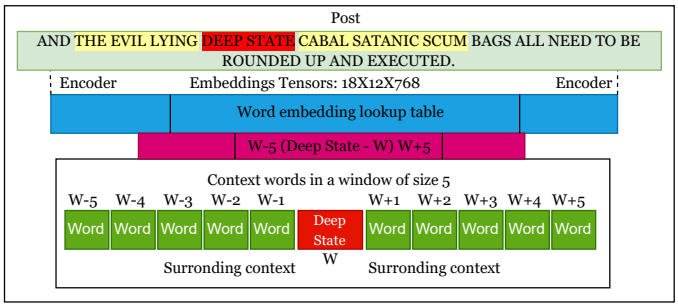

Fig. 4. Post-truncate embedding approach for a window of size 5.

vides complete context for each post.[20] Once this embedding is constructed, we follow the same procedure described in Section V-B2a except for the fact that we now extract word-level contextual embeddings from the lookup table (see Figure 4). The advantage of this approach over the previous one is that it builds more informed embeddings given its use of a complete rather than partial context. Please note that there are three additional differences between Approaches 1 and 2: in Approach 1, we used context window sizes between 5 and 14 while in Approach 2, we used context window sizes between 1 and 10. That is because a window of 1 word does not convey much information in Approach 1 whereas it does in Approach 2. As a result, we started at size 5 in Approach 1, and 1 in Approach 2, and used 10 different window sizes in each case. Furthermore, in Approach 2, the embeddings are generated by averaging the final encoder layer of BERT rather than using the pooled layer since that yielded better results. Finally, the optimal value for m in Approach 2 was 9 rather than 7.

[20]We assume that each word in the post has a token id in Bert's vocabulary.

VI. Results and Discussion

The purpose of our study was to design a methodology for extracting emerging coded antisemitic terminology from online posts appearing on social media platforms often used by extremist groups. We proposed a pipeline to implement this methodology and instantiated it using two different components in each phase, which yielded four different solutions. To assess the performance of our solutions, we created a gold standard and tested our results according to it.

A. Construction of a gold standard:

Our gold standard uses two complementary methodologies. One for the terms already familiar to the community that fights antisemitism, and the other, for the terms unknown or not yet catalogued by that community.[21]

Known Terms For the first category, we pitted the terms discovered by our approaches against three existing sources: the Institute for Curriculum Services' Glossary spanning the history of European Antisemitism, which we took in its entirety; the American Jewish Committee "Translate Hate" glossary which we also used in its entirety (prior to its recent expansion from 46 to 70 terms) and portions of the Glossary of Terms and Acronyms constructed by the R2Pris project on Radicalization and violent extremism. If the term was found in one of these sources (either as a defined term or inside a definition), it was categorized as a *known term*. [22]

New Terms The new terms are the terms that do not appear in the glossaries just mentioned and that need to be manually verified through an internet search. We used a systematic procedure based on three rules to assign ground labels to new terms. The details of our procedure are available in the GitHub repository referenced in Footnote 15.

Qualitative evaluation We conducted two types of qualitative evaluation. The first one simply consisted of observing the terms extracted by the approach to assess whether they made sense when taken out of context. The second one can be thought of as a sanity check. For terms extracted and labeled as either antisemitic or not, we went back to the the posts from which the term was extracted to assess whether, within the context of the post, it was used in an antisemitic way or not. Though we do not use these qualitative assessments in our quantitative evaluation, we show examples of the different situations that arose in terms of agreement or disagreement between our system and our gold standard. Specifically, coded

[21]The seed words used in this study represent only a small subset of the already known coded antisemitic terms. Therefore, some of the emergent terms discovered by our system are emergent vis-a-vis the system's knowledge but not vis-a-vis the broader current knowledge. Discovering terms known to the community but not known a-priori by the system constitutes a useful proof of concept. The discovery of terms not currently known by the community constitutes an added demonstration of the worth of the approach.

[22]The sources we used can be found at the following websites: https://bit.ly/45kEtYB; https://bit.ly/3MIjKpt; and http://www.r2pris.org/glossary.html

terms that are assessed as wrongly classified as antisemitic by our quantitative evaluation method, but believed to be potentially antisemitic based on our qualitative approach are indicated as potential terms.

B. Results

Quantitative Results: We tested the four different versions of our proposed pipeline, by combining Approaches 1 and 2 proposed for trending term extraction with Approaches 1 and 2 proposed for term embedding. These combinations resulted in solutions 1-1, 1-2, 2-1, and 2-2. Table III lists the results obtained by each of these solutions. The results were obtained using our gold standard labels. Approach 2-2 stands out as the absolute winner, although the results for all four solutions, including 2-2, show a higher level of recall than precision. Future work will attempt to improve all these metrics' scores, with a focus on precision so as not to unduly label terms as antisemitic when they are, in fact, benign. When comparing the numbers in Table III, it is important to note that the number and type of terms retrieved differ between the two term extraction processes, Approaches 1 and 2. While Approach 1 extracted 52 terms of which only 7 were truly antisemitic, Approach 2 extracted 94 of which 29 were truly antisemitic. The recall of 1 obtained by the two Approach 1-based methods, thus means that both Embedding Approaches 1 and 2 were able to identify these 7 antisemitic terms. Their low level of precision, however, suggests that they are too liberal in their labeling of terms as antisemitic.

Qualitative Results: Our qualitative evaluation was applied to the version of our pipeline that obtained the best results: Solution 2-2. Table IV shows some of the terms extracted by that version. The terms in red correspond to terms incorrectly classified as antisemitic with no good explanation; those in black are correctly classified as antisemitic as they correspond to our *Known Terms* in our gold standard evaluation protocol; those in blue were verified to be antisemitic as they correspond to our *New Terms* in our gold standard evaluation protocol; and those in purple were incorrectly classified as antisemitic by our gold standard evaluation protocol, although the context in which they arise is clearly antisemitic and we believe that they qualify as coded antisemitic terms even though our strict gold standard protocol does not assess them to be. As discussed before, we call these terms *Potential Terms (in an antisemitic context)*. The appendix illustrates the use of some of these terms by listing the posts in which they occur and discussing the relationship between their classification and the post itself.

C. Discussion

Though we know that our approach could still be refined, we note that the results obtained by the most successful version of our pipeline are encouraging, suggesting the viability of automatic emergent antisemitic coded terms extraction, and, by extension the viability of extracting emergent coded terminology for any kind of hatred. The qualitative analysis we conducted suggests that the terms identified by our approach are, usually, warranted as the context shown in the posts (see appendix) attests to the antisemitic nature of the discourse. The cases where they aren't are very useful as they point to the errors made by the system and will help us improve our results in the future. We believe that our approach could have important practical uses. We already demonstrated the usefulness of coded terms in automatic hate speech detection. However, that application was time-bound. Yet, we believe that hate speech on social media evolves continuously. Creating a system that constantly analyzes new posts, extracts new terminology from them, and feeds that information to a hate speech detection system is likely to enhance the quality of hate speech detection continuously.

VII. Conclusion

This paper proposes an approach for detecting the emergence of new antisemitic coded terminology which offers a valuable resource in combating online antisemitism and contributes to the ongoing efforts to create safer and more inclusive online spaces. We achieve an accuracy of 80% and F-Score of 72% in extracting antisemitic terms using this approach which relies on NLP techniques including POS tagging, TF-IDF, and fined-tuned large language models such as BERT. In the future, we intend to refine our semantic similarity technique by exploring other deep learning and large language model approaches and their various parameter combinations. Similarly, we will experiment with different types of text pre-processing approaches to deal specifically with hate-speech and social media text. This will be done in the context of a lifelong-learning setting where the trending terms discovered will be used as input to the data scraping component in the following iteration. We also intend to create a more user-friendly version that will be convenient for people working in this space. Finally, our goal is to extend this study to hateful terminology against other minority groups.

References

[1] M. Chandra, D. R. Pailla, H. Bhatia, A. J. Sanchawala, M. Gupta, M. Shrivastava, and P. Kumaraguru. "subverting the jewtocracy": Online antisemitism detection using multimodal deep learning. *Proceedings of the 13th ACM Web Science Conference 2021*, 2021.

[2] N. A. Cloutier and N. Japkowicz. Fine-tuned generative llm oversampling can improve performance over traditional techniques on multiclass imbalanced text classification. *IEEE Confernece on Big Data*, 2023.

[3] P. Fortuna and S. Nunes. A survey on automatic detection of hate speech in text. *ACM Computing Surveys (CSUR)*, 51:1 – 30, 2018.

[4] A. M. Founta, D. Chatzakou, N. Kourtellis, J. Blackburn, A. Vakali, and I. Leontiadis. A unified deep learning architecture for abuse detection. In *Proceedings of the 10th ACM conference on web science*, pages 105–114, 2019.

[5] B. Gambäck and U. K. Sikdar. Using convolutional neural networks to classify hate-speech. In *Proceedings of the first workshop on abusive language online*, pages 85–90, 2017.

[6] A. Glazkova. A comparison of text preprocessing techniques for hate and offensive speech detection in twitter. *Social Network Analysis and Mining*, 13:1–28, 2023.

[7] M. S. Jahan and M. Oussalah. A systematic review of hate speech automatic detection using natural language processing. *Neurocomputing*, 546:126232, 2021.

[8] G. Jikeli, D. Cavar, and D. Miehling. Annotating antisemitic online content. towards an applicable definition of antisemitism. *arXiv preprint arXiv:1910.01214*, 2019.

TABLE III
ACCURACY, PRECISION, RECALL AND F-SCORE USING THE FOUR VERSIONS OF OUR PIPELINE.

Solution	Accuracy	Precision	Recall	F-score
1-1	0.79	0.39	1	0.56
1-2	0.77	0.37	1	0.54
2-1	0.68	0.49	0.59	0.53
2-2	**0.80**	**0.63**	**0.83**	**0.72**

TABLE IV
LIST OF TRENDING TERMS THAT ARE PREDICTED ANTISEMITIC BY VERSION 2-2 OF THE PIPELINE.

False Positives		Known Terms		New Terms		Potential Terms	
plain sight	german people	white genocide	deep state	FEMA camps	klaus schwab	end game	world war
new york city	big part	interest groups	kazarian mafia	central bank	national socialist	western civilization	democrat party

[9] G. Jikeli, S. Karali, D. Miehling, and K. Soemer. Antisemitic messages? a guide to high-quality annotation and a labeled dataset of tweets. *ArXiv*, abs/2304.14599, 2023.

[10] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] E. Loper and S. Bird. Nltk: The natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[12] W. Melillo, J. Emami, S. Guarinos, D. Kikkisetti, M. Klein, L. Liubovitch, R. Ul Mustafa, and N. Japkowicz. Seeking optimal human/machine collaborative practice in antisemitic terminology detection. *Proceedings of the 4th IEEE Conference on Digital Platforms and Societal Harms (DPSH)*, 2024.

[13] R. U. Mustafa, M. S. Nawaz, J. Farzund, M. Lali, B. Shahzad, and P. Viger. Early detection of controversial urdu speeches from social media. *Data Sci. Pattern Recognit.*, 1(2):26–42, 2017.

[14] S. Parker and D. Ruths. Is hate speech detection the solution the world wants? *Proceedings of the National Academy of Sciences of the United States of America*, 120, 2023.

[15] F. Poletto, V. Basile, M. Sanguinetti, C. Bosco, and V. Patti. Resources and benchmark corpora for hate speech detection: a systematic review. *Language Resources and Evaluation*, 55:477 – 523, 2020.

[16] J. Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242:1, pages 29–48. Citeseer, 2003.

[17] A. Schmidt and M. Wiegand. A survey on hate speech detection using natural language processing. In *SocialNLP@EACL*, 2017.

[18] M. Schwarz-Friesel and J. Reinharz. *Inside the antisemitic mind: the language of Jew-Hatred in contemporary Germany*. Brandeis University Press, 2017.

[19] J. Serra, I. Leontiadis, D. Spathis, G. Stringhini, J. Blackburn, and A. Vakali. Class-based prediction errors to detect hate speech with out-of-vocabulary words. In *Proceedings of the first workshop on abusive language online*, pages 36–40, 2017.

[20] G. Wiedemann, S. M. Yimam, and C. Biemann. Uhh-lt & lt2 at semeval-2020 task 12: Fine-tuning of pre-trained transformer networks for offensive language detection. *ArXiv*, abs/2004.11493, 2020.

[21] S. Zannettou, J. Finkelstein, B. Bradlyn, and J. Blackburn. A quantitative approach to understanding online antisemitism. In *Proceedings of the International AAAI conference on Web and Social Media*, volume 14, pages 786–797, 2020.

APPENDIX

The purpose of this appendix is to look at a few specific posts, the three listed below, and explain the different facets of our quantitative and qualitative evaluation approach. The colors used in the text correspond to those used in Table IV.

- "(...) We must stop cultural Marxism, Globohomo, Jewish supremacy, **white genocide**, usury, giving away our resources to savages, etc. "
- "FEMA is not a good thing! **FEMA camps** are concentration camps. **FEMA camps** are the **end game** of the **New World Order**"
- "all turds need to be deported from the West. turds are brown MENA sunni muslim garbage. they are a **big part** of the non-white invasion. (...) turds are also zionists and turdistan is a base for israeli ops. imagine sympathizing with these zio-muslim invaders."

We now comment on the terminology extracted by our 2-2 solution in view of the context in which it occurred. In particular, we show the context for the term **white genocide** which has an entry in the antisemitic glossary compilation described earlier. In particular, **white genocide**, refers to a conspiracy theory rooted in white supremacist ideology, claiming that there is an intentional effort by Jews to destroy the white race through immigration, mixed-racial marriage, LGBTQ+ identification, etc. The above list of posts also shows an instance of a new term —**FEMA camps**. This corresponds to a conspiracy theory where FEMA is believed to plan the incarceration and possible execution of US citizens in favor of the establishment of a "New World Order", one of our seed words which often refers to the establishment of a new form of government controlled by a "Jewish elite". On the other hand, during the process of extracting coded antisemitic terms, some terms were labeled as antisemitic despite the fact that they do not appear in our gold standard compilation. In certain cases, that represents an outright mistake like in the case of **big part** in the third post, where the context is certainly racist, but not specifically antisemitic, though antisemitism is part of the post, but in other situations, a case could be made for an antisemitic label. For example, our approach predicts **end game** as a coded antisemitic term, even though we did not find any reason for it based on our gold standard evaluation protocol. A look at the post in which the term appeared, though, helps us understand how the antisemitic context of the post that includes the terms "concentration camps" and "new world order" led the system to label it as such. We conclude that, in such cases, our approach may be extracting correct terms according to the context, but that the current gold standard protocol we used is too strict to recognize it. We call these *Potential Term (in an antisemitic context)*.