Multifaceted NLP Analysis of Hate Speech And Kinetic Action Descriptions Online

by

Bdour Al-Zeer

B.Sc. German Jordanian University, 2017

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Master of Science

in the
Department of Computing Science
Faculty of Applied Sciences

SIMON FRASER UNIVERSITY
Fall 2019

Approval

Name:	**Bdour Al-Zeer**
Degree:	**Master of Science (Computing Science)**
Title:	**Multifaceted NLP Analysis of Hate Speech And Kinetic Action Descriptions Online**
Examining Committee:	**Chair:** Parmit Chilana Assistant Professor, Computing Science
	Fred Popowich Professor, Computing Science Senior Supervisor
	Anoop Sarkar Professor, Computing Science Supervisor
	Dave Campbell Professor, Mathematics and Statistics External Examiner, Carleton University
Date Defended:	**Dec 16, 2019**

Abstract

Despite the many great advantages of social media and online forums in bringing people, communities, and groups together, other problems have emerged when using these sites including hate speech and abusive behavior online. Unfortunately, these platforms can be used as spaces to bully, harass, assault or even plan to carry on a kinetic action against others. Most of the data that comes from these sources is noisy, unstructured and unlabelled, which makes designing supervised classifiers a task that requires a lot of human effort for labeling and going through the data to determine the severity of toxicity in it. Also, the human toll of working with this data may include negative psychological effects on the person after reading a potentially large amount of data. For these reasons, our goal is to provide a framework to help perform an exploration of such unstructured data to be able to determine the important topics, features, sentiment, and entities involved without the need to manually read all the text, including providing the capability for the automatic redaction of toxic terminology. The net result would be an improved environment and exposure for people that need to analyze this data to explore these documents and identify documents of interest in a less harmful way. We use different state-of-the-art natural language and machine learning techniques to design a pipeline that takes in unstructured noisy data and converts it into actionable structured data. First, we pre-process and clean the data, performing tokenization and stemming. Second, we perform LDA topic modeling to cluster the data such that every cluster represents a topic. After clustering, we want to identify the entities involved in each document so we perform Named Entity Recognition NER to get the entities. Then, we do keywords list expansion on a small seed list of kinetic actions keywords - which was provided to us by subject matters- and use this list to perform filtered part of speech tagging POS to identify the documents that contain these keywords as Verbs (actions) and the entities that occurred as Objects in them. We also add another layer of detail by feeding the data into a state-of-the-art machine learning hate speech classifier that was trained on a huge set of labeled Twitter data to classify the data into three classes (Hate speech, Offensive Language, Neither). As a means of evaluation, we randomly sampled 300 records of the classifier results on our data and had human experts classify them. We also design a simple and modifiable scoring scheme that combines all the features of the multidimensional analysis and returns a score that can be used as a filtering metric to perform information retrieval on the documents, thus prioritizing those that require human intervention. We additionally perform visualization on the results to make it easier to comprehend and analyze by experts. We then provide an evaluation of the resulting system that incorporates a range of objective and subjective criteria.

Keywords: Query expansion, topic modeling, hate speech, kinetic actions, multi-dimensional analysis, information extraction and retrieval.

Dedication

To my beloved parents Abed and Bushra,
and my dear brother Bader

Acknowledgements

First and foremost, it is with immense gratitude that I thank my supervisor Dr. Fred Popowich for his amazing guidance, full support, encouragement, and patience during my studies at SFU. I especially thank him for the incredible efforts he put in the training and guiding me throughout the way. Beside the many scientific, academic, industrial gains that I got by working with Fred, what I really appreciate the most is the love he planted in me for research and learning and how he mentored me in a friendly and stress-free style which enabled me to have a healthy way of growth both as a student, researcher, and computer scientist.

I would like to take the opportunity to deeply thank the Canadian Association for Security and Intelligence Studies - Vancouver, CASIS, whom without their support, contribution, data, and resources this thesis would not have been as insightful and valuable. The data they voluntary shared with us enabled this study to have the proper material and substantial significance. I would like also to thank Dr. Anoop Sarkar and Dr. Steven Bergner for their valuable advice and sharp feedback in this thesis. Their insightful on-point comments helped me a lot refine and improve my work.

Last but not least, I whole-heartedly thank my loving family especially my Mom and Dad for all their support, love and care that they showered me with throughout not only this degree but during my whole life, making you proud was always on my mind. A sincere thanks to all my close friends in Canada, Jordan, and Germany, your words of encouragement deeply meant a lot to me and powered me throughout the past two years. So to everyone who supported me by either a technical help, academical advice, a warm smile, a sweet gesture or a sincere prayer, I say thank you to each and every single one of you. I will be forever grateful for your love.

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Internet and Social Media enabled effortless and instantaneous communication between users around the world. The accessibility of the Internet to anyone anytime and for a user to be able to share any sort of content with a huge number of users changed the way humans communicate and operate in the past two decades. The positive side of such fast technology is the major exploitation in all fields and especially in data publication and how social media platforms — in particular — leveraged freedom of speech to everyone. However, on the other hand, not all data shared is of a good nature and safe. A large portion of the content online is abusive and hateful as unfortunately, some individuals adopt abusive behavior online. This sort of activity resulted in discussions on the immense need to balance between freedom of speech and similar harmful conduct that may have severe effects.

According to a recent report on online harassment by the Pew Research Center [18] 40 percent of American adults have experienced online abuse of which 18 percent have faced severe forms of harassment, e.g., that of sexual nature and 66 percent have witnessed it happening to other users. Meanwhile, a youth campaign called *No Hate Speech Move* [2] led by the Council of Europe Youth Department was launched to combat hate speech and promote human rights online. Also, UNESCO published a study [23] under the title "Countering Online Hate Speech" that provides a global overview of the dynamics characterizing hate speech online and some of the measures that have been adopted to counteract and mitigate it.

Furthermore, another important reason why it is a critical problem is the evident link between online hate speech and real-world hate crimes [33] so, it is rather urgent to provide tools and means to try and prevent such an escalation from online hate into real-world kinetic hate crimes, in chapter 3.1 we provide our own definition of the term kinetic action. All of these state that toxicity and abuse on the web are globally recognized as a serious problem of our time and led to an increase in combined research and development efforts to address the issue. Recently, new research in 2019 on the link between online hate speech and actual hate crimes in the real-world by Williams et al. confirms that there is an association between the two [64].

Online abuse and misconduct comes in numerous forms from inciting hate speech, misinformation and fake news circulation, bullying, harassment, threats or even to plan to carry out a kinetic action against a particular person or a group. All forms of abusive behavior online could potentially leave deep psychological, emotional and mental effects on both the targeted parties and the domain experts whose jobs involve dealing and analyzing such unpleasant data. Most of the data that come from these online sources is noisy, unstructured and unlabelled which makes designing supervised classifiers or analyzing the data, a task that requires a lot of human effort for labeling and going through the data to determine the severity of toxicity in it.

That is why in this work, we first aim at providing a comprehensive framework for the exploration of such unstructured data providing the capability for the automatic redaction of toxic terminology. And using existing natural language processing (NLP) and machine learning techniques to provide insight into such complex data to identify the targets of hate speech and to determine the important topics, features, sentiment, and entities involved without the need for people to read all the text.

1.2 Thesis Contribution

This thesis aims to expose insight into fighting hate speech and abusive content in online environments -like social media platforms- using NLP techniques in general and specifically from a visualization point of view. In particular, we develop an approach that will help solve some of the challenges mentioned in section 1.1. We use this approach to conduct an analysis to gain insights on the importance of creating such a comprehensive multidimensional framework and demonstrate the need for it in the field of abusive textual content visualization, exploration, and retrieval.

Our approach is directed towards making navigating, knowledge retrieval, structuring, analyzing and visualizing data of opprobrious nature an easier task with increased time efficiency and decreased interpersonal disadvantages for domain experts.

1.3 Thesis Outline

The chapters in this thesis are organized as follows:

Chapter 1 Introduces the topic of study and motivation behind it. It goes over the challenges faced in the field of abuse and hate speech on the Internet.

Chapter 2 Starts by exploring related work in the field of hate speech and abuse online. Then it explains to the reader the background of all the natural language processing techniques and concepts that we made use of in our work.

Chapter 3 Describes our multifaceted approach, going over the high-level details of every step. The chapter then goes on to describe the data used in this study and why we chose it.

Chapter 4 Details of our system implementation, it goes over every phase of the introduced multifaceted approach and how all steps support the analysis required by domain experts.

Chapter 5 Details the visualization component of our approach. First, we go over the idea of accessing the data in a multidimensional style. Then the chapter explains the web application designed to showcase our approach through a proof of concept study.

Chapter 6 Evaluates the findings of our study and details a domain expert use-case test evaluation of the usability and application of such approach.

Chapter 7 Concludes the work done and ends with several proposals for future work in this area.

Chapter 2

Background and Related Work

This chapter starts by exploring some related work in the field of hate speech and abuse online. Then, it explains to the reader the background of all the natural language processing techniques and concepts that we made use of in this study.

2.1 Related Work

Although substantial research and attention were given to the problem of hate speech and abuse on the internet, there remains a lot to do to establish a safe online environment and a thorough understanding of the possible serious side-effects such behaviors might lead to. In this work, we are addressing the problem of exposure redaction to opprobrious content for a subject domain expert - someone whose job involves analyzing, going through and making real decisions based on this content. And to establish that, we came up with a framework that combines multifaceted and multidimensional representations of documents along with their summarization and features that were derived using natural language processing, lexical and semantic features. In this section, we highlight research in two main areas:

2.1.1 Work in Hate Speech Area

Over the past several years, the attention directed towards hate speech and abuse online has increased notably due to reasons we highlighted before, the majority of work in this area are mainly supervised or semi-supervised learning detection and classification systems whether trained via a monolingual dataset like for instance [4], [38] and [16] or a multilingual dataset like [45], [25], [57] and [6]. Currently, one of the best performing works for classifying hate speech combines semantic word embeddings obtained via recurrent neural networks along with decision trees [4]. Another notable work utilizes recurrent neural networks along with an attention mechanism for user comments moderation [47].

There is also the work that focuses on generally understanding the problem of hate speech online like the one proposed by Faris et al. in [19]. Such work focuses on understanding the research done in the area of harmful content online. Other research efforts address the problem by designing language monitors or training classifiers and detection systems using rule-based linguistic systems and machine learning. There is also the research that attends to the problem from one or multiple dimensions like systems built to extract events linking to hate crimes representation via training a model on a corpus of news articles and using it to detect hate incidents [15]. Or the neural networks designed to utilizes the emotions in a document along (attention mechanism) with an architecture that takes into account the local and sequential information of text for abuse detection [53]. Recent work proposed utilizing user-profiling (community-based) features to detect hate speech on social media building on previous work that states "hateful content tends to come from users who share a set of common stereotypes and form communities around them" [38].

In section 2.3, we provide extra insights into the problem of hate speech classification and its background.

2.1.2 Work with similar visualization approaches

In this subsection, we will highlight some of the previous work that applied relatively similar representation and information retrieval approaches. In other words, work that adopted multifaceted navigation techniques but in different domains. We will highlight only two of such works to explain the concept of it, additionally, we will provide more insights into the idea of multifaceted navigation in section 5.1.1.

Back in 1997, the idea of Multidimensionality and Multifaceted was also explained by Kyo Kageura. In the context of concept systems in terminological representations, Multidimensionality has been explained as the case when a concept can be classified with respect to many different facets and set of characteristics. And a concept was defined as: "by a concept, we basically mean those units of thought actually or potentially represented by terms" [32]. Which is a similar principle to our framework since it is also approaching the problem from multiple dimensions and facets to describe the same document. The main strength of such multidimensional/multifaceted systems is the ability to have a complete set of characteristics from various angles and not restrict it to a single context or dimension. So that at the end all of it complements the bigger picture of the comprehensive understanding of an entity — in our case its a document. Figure 2.1 explains the idea in a static conceptual organization space.

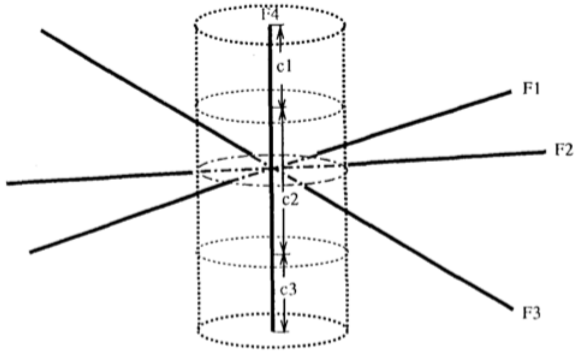


Figure 1: *A Subspace within a Static Conceptual Organization*

Figure 2.1: In the illustration above every line represents a facet or a dimension that all describes a single concept but from a different angle. Source: Handbook of terminology management: Basic aspects of terminology management [32].

Moreover, in 2010, Cao et al. proposed an approach to perform document visualization via multiple facets of information reasoning that documents often encompass various aspects of information. They referred to the technique as FacetAtlas and defined it as "a multifaceted visualization technique for visually analyzing rich text corpora, FacetAtlas combines search technology with advanced visual analytical tools to convey both global and local patterns simultaneously". This research is interesting to us because we are also proposing to analyze the documents from different angles via different methods. Mainly the idea of having multiple aspects of information to describe and summarize a single document. Figure below 2.2 illustrates their model idea [12].

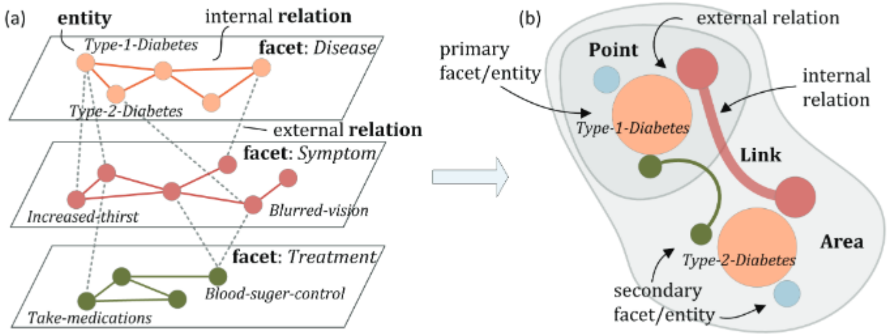


Fig. 2. (a) The FacetAtlas multifaceted entity-relational data model. Concepts in a complex text corpus are transformed into *facets*, *entities* and *relations*. (b) The data model is visually encoded using a spatial arrangement of color-coded nodes and edges.

Figure 2.2: This figure illustrates the idea of Multifaceted of having various aspects and categories to describe entities rather a single dimension. Source: [12].

2.2 Topic Modelling

With the immense increase of data creation, data sharing and data collecting that are happening every day, a need has emerged for better ways to access, organize and analyze all of this data — often unstructured data. What we usually mean by a topic model is a statistical and probabilistic model for discovering the hidden topics that a collection of documents talk about which helps in return with annotating these documents with respect to discovered topics and thus helps with understanding, providing insight, summarization, and organization of big collections of documents.

The concept of topic modeling has evolved to enable a better understanding of such large data collections. It was first described in 1998 by Papadimitriou et al. [46]. Later in 1999, another topic model called PLSA — which stands for probabilistic latent semantic analysis — was proposed by Hofmann [29]. After that, in 2003 the most popular topic model known these days as Latent Dirichlet Allocations (LDA) — which is a generalization from what [29] proposed — was developed by Blei et al. [8].

2.2.1 Latent Dirichlet Allocations - LDA

LDA as defined in [8] is "a generative probabilistic model for collections of discrete data such as text corpora, it is a three-level hierarchical Bayesian model, in which each item of a collection is modeled as a finite mixture over an underlying set of topics. Each topic is, in turn, modeled as an infinite mixture over an underlying set of topic probabilities. In the context of text modeling, the topic probabilities provide an explicit representation of a document".

In simple words, LDA assumes that every document contains a set of topics rather than a single topic and each topic is represented by a set of keywords drawn from the words in these documents - a corpus of text. LDA's main job is to identify these keywords in a way such that each topic is like a cluster of keywords along with the probability of how much each keyword is occurring in that topic cluster. Also, it is possible that the same keyword occurs in multiple documents but it would have a different distribution per document. Often, when topic modeling is performed well this set of keywords per topic helps with interpreting the discussed topic.

The question now is how LDA achieves that and how exactly it works. As per the model name latent — which means hidden — the model assumes a hidden layer of a number of topics that we do not know yet. This hidden layer represents the set of documents we have as our text corpus. Now the model goes on in a generative process by assigning keywords out of the words in all of the documents to each one of these assumed topics. The illustration in figure 2.3 describes this idea.

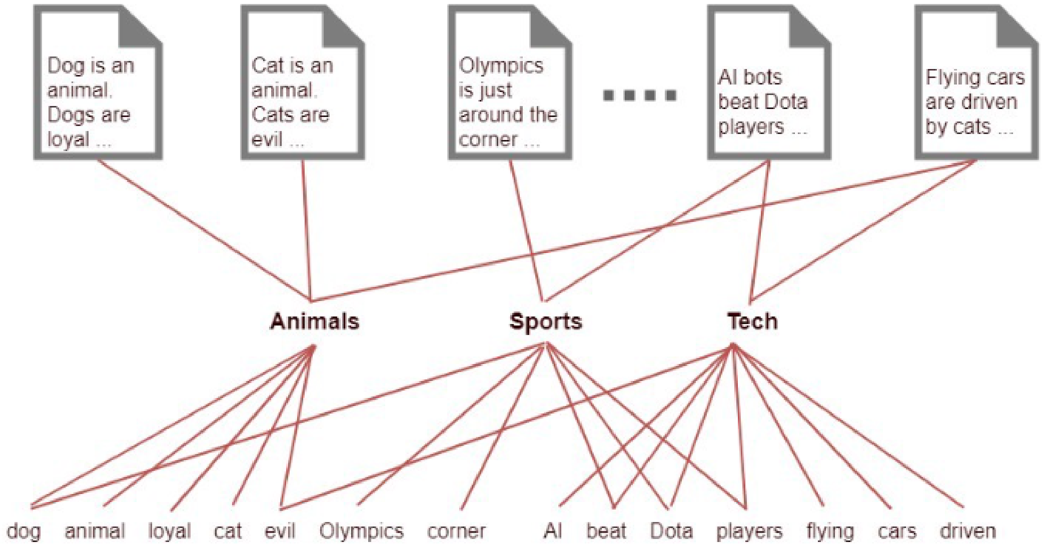
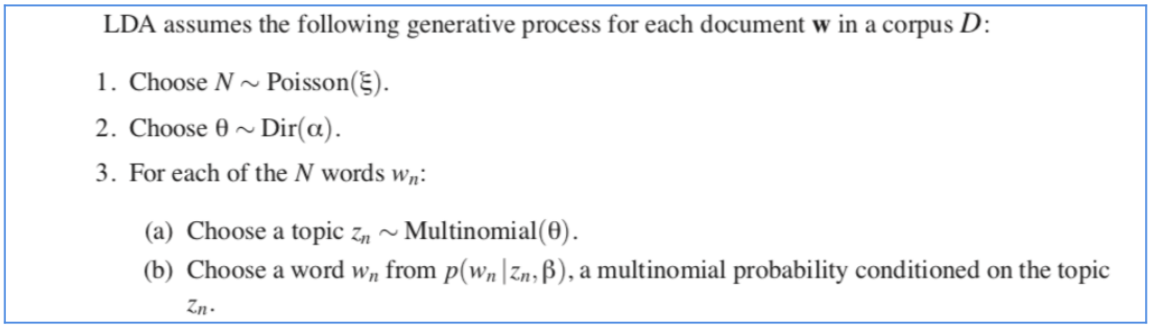

Dog is an animal. Dogs are loyal ...
Cat is an animal. Cats are evil ...
Olympics is just around the corner ...
....
AI bots beat Dota players ...
Flying cars are driven by cats ...
Animals
Sports
Tech
dog
animal
loyal
cat
evil
Olympics
corner
AI
beat
Dota
players
flying
cars
driven

Figure 2.3: Documents and words are formed by a set of topics. Source: [24].

As we said before, LDA is a generative model meaning that all documents are created in a word-by-word process by choosing a topic mixture such that we can infer the topics in a corpus in a reverse engineered way. The algorithm starts by picking k topics (chosen arbitrary before running the model) and then assigning a random topic for each word in each document in the corpus - referred to as the α, and after that in a generative process per document, improve the topic distributions for each document, α, and word distributions for all topics, β. After the model training finishes, the LDA result is a number of topic clusters such that for each keyword in these clusters there is a topic distribution that describes to which degree this keyword belongs to each topic (cluster). Of course, keywords in the same cluster have a close distribution relative to that cluster.

In more formal terms, an LDA model is usually defined by two main parameters, the first, α, is a prior estimation on topic probability per each document and the second, β, is the word-topic distribution for each topic. As illustrated in figure 2.4.



LDA assumes the following generative process for each document $\mathbf{w}$ in a corpus D:

1. Choose $N \sim \text{Poisson}(\xi)$.
2. Choose $\theta \sim \text{Dir}(\alpha)$.
3. For each of the N words w_n:
   (a) Choose a topic $z_n \sim \text{Multinomial}(\theta)$.
   (b) Choose a word w_n from $p(w_n \mid z_n, \beta)$, a multinomial probability conditioned on the topic z_n.

Figure 2.4: LDA as described by Blei et al. in [8]

Where,

- N: corresponds to the total number of words in all documents in corpus D,
- α: corresponds to the Document-Topic weight,
- β: corresponds to the Word-Topic weight,
- θ: corresponds to the distribution of all topics in document w,
- z: corresponds to the topic for the n-th word in document w.

2.3 Hate Speech Classification and Detection Using NLP

As the use of social media and other web-based platforms is increasing a lot every day worldwide, the amount of data on such platforms is also increasing. Not all of this content is appropriate with the amount of abuse and hate speech content increasing. Thus, a need has emerged to address this problem — as we will describe shortly — and to design methods to help detect, classify and monitor such bad content, especially, considering the profound psychological and lasting effects of it. For these reasons, the area of natural language processing has witnessed a lot of substantial research work to prevent and detect different types of abuse and hate speech on the web.

In this section, we will provide an overview of the most recent work and describe the methods used so far in this field. Since basic word filters and rule-based systems are not always enough to address the problem of hate speech because of the wide umbrella that this term might hold beneath it. As the question of what is considered hate speech can be affected by various angles, for example the targets of hate speech might be minorities, religion or political parties. Moreover, the occurrence of any real-time world events in a close time span to the message shared online, the type of language and words used, the context of which it was posted in and whether it can be considered as just an expression of users opinion under the freedom of speech or is it something different that might have harmful effects on others, would have also an impact. A plethora of questions and aspects empathized with the urge for developing better techniques such as language understanding and language processing methods that can incorporate these facets.

In section 2.3.1, firstly, we will start with the definition of hate speech online. Then, in section 2.3.2, we will describe the techniques that involved more linguistic and feature engineering approaches and after that, in section 2.3.3, we will explain the more recent approaches that relied more on neural networks and deep learning techniques.

2.3.1 Definition of Hate Speech Online

Due to the many and wide meanings that hate speech can have, an important step was to define it and to have a generally agreed-upon description of what is considered abuse and hate speech on the web and the many characteristics of it. That is why, most of the work done so far starts by providing a definition that they adopt throughout the work. However, having too many definitions might affect the overall progress of the problem and the ability to build upon previous work.

A system called Smokey for Automatic Recognition of Hostile Messages was developed in 1997 by Spertus and it is considered one of the earliest works in this field. It refers to hate speech as abusive messages, hostile messages or flames [59]. In 2012, work done to detect hate speech in online text by Warner et al. defines hate speech text as "abusive speech targeting specific group characteristics, such as ethnic origin, religion, gender, or sexual orientation" [63]. A similar definition was adopted by Burnap et al. in 2015, which interpreted hate speech as "written expressions of hateful and antagonistic sentiment toward a particular race, ethnicity, or religion" [11]. To the best of our knowledge, there is not yet an officially recognized definition of hate speech, it is rather conceptually agreed upon, that hate speech is any speech that targets disadvantaged groups in a way that may be harmful to these groups [62][31][16].

In this work, we will use the definition adopted by Davidson et al. in their work [16] in which they define it as "a language that is used to expresses hatred towards a targeted group or is intended to be derogatory, to humiliate, or to insult the members of the group. In extreme cases, this may also be language that threatens or incites violence".

2.3.2 Hate Speech Classification Background Work

In this section, we will briefly go over the most notable techniques and approaches that have been done in the past on the topic of abuse and hate speech online detection — to do so, we referred to a recent survey on this topic done in 2019 by Mishra et al. [39]. The work is categorized on which techniques each used, whether it was based on features engineering techniques or more recent ones that relied on neural networks and deep learning methods. In 2018, Van et al. did a study on "Challenges for toxic comment classification" [61] and found that a mix of both techniques (neural and non-neural) resulted in the best performance.

Features Engineering Techniques

The work on abuse and toxicity began quite a while ago. The system called "Smokey" relied on features engineering and a heuristic rule-based approach for detection [59]. The features were based on the syntax and semantics of each sentence. The linguistic features were designed for the task of flame detection (similar to what we call nowadays hateful or abusive messages). The author used a tree generator to determine these features and, also, in her work she discusses the other added rules and user-customized techniques to increase the overall accuracy.

Another work based on features engineering was done by Sood et al. in 2012 on profanity detection titled "Using Crowdsourcing to Improve Profanity Detection". In this work, the authors built a system that moves beyond previous list-based systems. They integrated bag of words features B.O.W and designed an SVM classifier that was trained on word bi-grams features [58]. More recent work was done by Salminen et al. in 2018 on the "Anatomy of Online Hate". The authors follow a features engineering approach that trains a linear SVM classifier using weighted n-grams to detect and classify hateful comments in the context of online news media, the model achieves an average F1 score of 0.79 using TF-IDF features. They also prepare and annotate manually 5,143 hateful expressions on YouTube and Facebook. In addition, they create a granular taxonomy that includes both types and targets of hateful comments [52].

Neural Network and Deep Learning Techniques

Now, moving to the most recent work that used neural networks and deep learning techniques. In 2015, Djuric et al. were the first to use neural networks to the problem of hate speech detection in online user comments. The authors use paragraph2vec to learn the distributed low-dimensional representations of comments and use these learned embeddings to train a logistic regression (LR) classifier [17]. After that in 2016, Nobata et al. worked on "Abusive language detection in online user content", building on the work done by Djuric et al. in 2015 [17]. The authors propose a supervised classification method to detect hate speech in online user comments via features that captures different aspects of the comments [42].

Later in 2017, Davidson et al. in "Automated hate speech detection and the problem of offensive language" also used neural networks to create a multi-class classifier however, they did not only aim to detect hateful content but they are also interested to build "a fine-grained classification model to distinguish between sentences containing hate speech, only offensive language, and those with neither". To achieve that, they use crowd-sourcing to label a sample of these tweets into three categories and use this data as training data for a logistic regression model with L2 regularization [16]. Since they trained their model on social media hateful and offensive data and their model achieved relatively very good accuracy we decided to use their model as the hate speech classifier component in our analysis. We will provide more details on this work in 4.3.

2.4 Information Extraction - IE

Information Extraction is selection of passages from documents whether they are structured or unstructured documents. Extracted information could be entities, attributes and the relationships between these entities as in the source document. Dale et al. define Information Extraction in their "Handbook of Natural Language Processing" book as: any process that selectively structures and combines data that is found, explicitly stated or implied, in one or more texts. The final output of the extraction process varies; in every case; however, it can be transformed such that it populates some type of database. Information analysts working long term on specific tasks already carry out IE manually with the express goal of database creation" [14].

The main aim of IE is to obtain structured information and their proper representations in the format of useful knowledge that is derived out of the source data. Or as Singh et al. in 2018 put it: " the goal of IE is to extract salient facts about pre-specified types of events, entities, or relationships, in order to build more meaningful, rich representations of their semantic content, which can be used to populate databases that provide more structured input" [56].

IE has various sub-tasks that all serve in the bigger goal of extracting useful knowledge from documents according to specific criteria. Some of these sub-tasks are:

- Named Entity Recognition: the task of identifying the involved entities in a document, paragraph, sentence or a sum of a text corpus.
- Part of Speech Tagging: the task of assigning linguistic lexical labels for words in a sentence such that it takes into account both the context of each word and the lexical meaning of it.
- Query Expansion: the task of adding more semantically or contextually similar words to a given query to reformulate it in a way that would increase the chances of finding more documents that correspond to the original query.
- Named Entity Linking: the task of assigning a unique identity to entities mentioned in text, like for example locations or famous figures.
- Relation Extraction: the task of extracting semantic relationships between entities - like born in, son of or married to - from a text.

In the following sections, we will briefly go over Part of Speech Tagging, Named Entity Recognition and Query Expansion to provide some background information since we used these techniques in our pipeline and analysis. The remaining techniques might be interesting to explore as additional future work and development on our approach.

2.4.1 Part of Speech Tagging - POS

Let us first start with the definition of part-of-speech (POS) tagging, which is the process of assigning linguistic lexical labels for words in a sentence such that it takes into account both the context of each word and the lexical meaning of it. These labels are called POS tags and an example of such tags are the labels (Verb, Adjective, Noun, ..., etc). For example, the sentence *"I like to play football I hated it in my childhood though"* would be tagged as shown in figure 2.3.

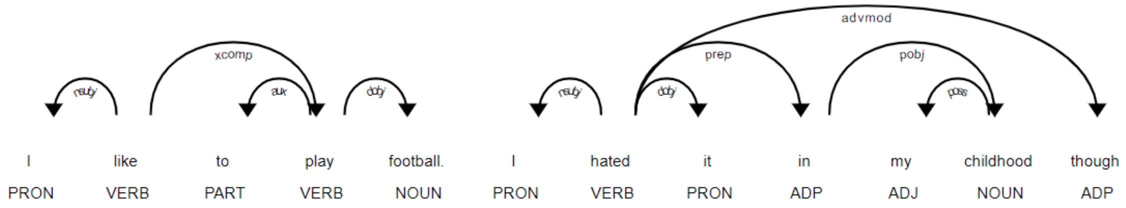


Figure 2.5: Example of POS and NER by Stackabuse. Source: [60]

Performing the annotation tagging manually can be an expensive and a labor-intensive human process. Currently it is done mostly via automatic annotation using a tool called a POS tagger. There are two main categories that the methods and algorithms for POS tagging:

- **Rule based approaches**: employ language specific rules and algorithms to perform POS-tagging. A well known English language POS-tagger in this group is E. Brill's tagger, from 1993, which automatically acquires its rules via supervised learning and a transformation-based process. It tags with an accuracy comparable to stochastic taggers [10]. The downside of this approach is it can be expensive sometimes in the sense that it needs a huge set of annotated data for a particular language and this may require lots of experts' knowledge and human effort to build.

- **Stochastic and statistical approaches**: are approaches that depend on hidden Markov models, decision trees models, support vector machines (SVM), maximum entropy classifier and lately using machine learning and deep learning methods [56]. A full comparison study on the performance of different methods in this group is provided in depth at the ACL Wiki [1]

[1] https://aclweb.org/aclwiki/POS_Tagging_(State_of_the_art)

2.4.2 Named Entities Recognition - NER

Now, let us move to one of the most important tasks in IE in particular, the task of Named Entities Recognition (NER), sometimes referred to as Named Entities Disambiguation or Named Entity Linking or Named Entity Normalization. NER is the process of identifying the involved entities in a document, paragraph, sentence or collection of texts. Those entities could be the persons mentioned, organizations, locations or geopolitical entities. It simply assigns a tag that indicates what type of entities this word (token) — sometimes more than one word — represents. An example is shown in Figure 2.6:

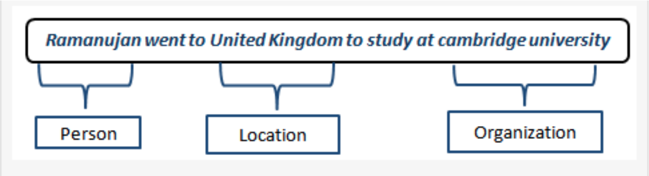

Ramanujan went to United Kingdom to study at cambridge university
Person
Location
Organization

Figure 2.6: NER Example by ListenData. Source: [43]

The history of research in NER started in 1991 by Lisa F. Rau with a research paper that proposes a system designed to extract and recognize a company name on a financial news corpus using a rule-based and heuristic approach and corpus analysis [49]. In 1999, a supervised learning approach was proposed by Bikel et al. in their work "An Algorithm that Learns What's in a Name", where they used a Hidden Markov Model to perform an NER task - identify names, dates, times, and numerical quantities [7]. Later, other works which also relied on Supervised Learning and Machine Learning techniques was proposed, such as approaches that used Decision Trees [55] or Maximum Entropy Models [9] or Support Vector Machines [3]. We referred to a NER survey by Nadeau et al. in 2007 for our NER summary [40], which provides more in-depth background knowledge.

Recently, a new state of the art is reported using a neural network approach by Facebook Research AI with an F1 score of 93.5 in the CoNLL 2003 NER task - which consists of newswire text from the Reuters RCV1 corpus tagged with four different entity types (PER, LOC, ORG, MISC) [5]. A summary of current state-of-art in the NER field is explained on the NLP progress site [51].

2.4.3 Query Expansion - QE

In information retrieval systems, Query Expansion and Query Understanding are integral tasks to have a successful information retrieval system. QE is defined as the task of adding more semantically or contextually similar words to a given query to reformulate it in a way that would increase the chances of finding more documents that correspond to the original query (better retrieval). An example of QE would be if a user query was *"Boeing*

Aircraft", the expanded query might look like *"Plane, Flight, Boeing, Airbus, manufacturing companies,..., etc"*.

In the literature, one of the earliest works that proposed a work to perform QE was done in 1960 by Maron et al., who proposed an novel automatic technique for literature indexing and searching [37]. Recently, more advanced techniques were proposed, they are methods that are ontology-based which follow semantic or syntactical or a mix of both methods [66]. Other recent automatic QE methods that mainly rely on local and global document analysis. Carpineto et al. in 2012 wrote a survey on such approaches [13].

There are many methods to perform QE that depend on thesaurus resources to find the synonyms, hyponyms, hypernyms of a word or to correct misspellings of the words in a query. Such resources include for example WordNet or Dictionaries. There are also other tools such as neural networks and Word Embeddings. Word embeddings are learned representations using language modeling and deep learning techniques for text such that words of similar meaning have a similar representation — real number vectors.

Recently, in 2016, state-of-the-art performance was reported by Nie et al., who proposed in "a novel method to improve the performance of code search algorithms" a query expansion based on crowd knowledge [41]. It reported an improvement over the previous state-of-the-art methods that did query expansion via WordNet for an effective code search by Lu et al. in 2015 [35]. Please refer to [66] [13] [44] for more in-depth surveys on Query Expansion background and current approaches.

Chapter 3

Approach

3.1 What is our definition of kinetic hate actions?

The word "Kinetic" itself as defined in the Cambridge English dictionary means "involving or producing movement". In our proof-of-concept study, we assume the following definition of Online Kinetic Hate Action: *it is any action or crime against a person or a property that was motivated or inspired by any sort of online content that is believed to be hate speech towards a certain individual, group or minorities.* The context of this definition is around any behavior or action that revolves around harmful acts carried on by individuals as a result of shared online content of hateful nature.

3.2 What is a Multifaceted Approach?

As mentioned previously, usually online data that contains such abusive and hateful content is accompanied by many linguistic and psychological challenges. The linguistic ones are for example the misspellings, slang language, and sarcasm. And the psychological ones are for instance the interpersonal, morale and mental disadvantages that might affect a human reader after exposure to disturbing content, especially when such exposure occurs in big data volumes.

We explore the data on multiple levels and dimensions, hence we refer to it as a Multifaceted Approach - MFA. In simple yet effective steps MFA achieves a model to first enable human interaction decrease with abusive and obnoxious content and second returns a ranked list of documents based on an embedded and modifiable scoring scheme that integrates together all the aspects of the framework. Our approach incorporates all the analysis facets results together.

Here, we are going to only highlight the *MFA* steps since we will provide an in-depth description in later chapters. The analysis starts by first performing topic modeling on the original unstructured data to cluster the data into the main topics in which every cluster represents a topic and each document is allocated into the cluster which it represents the most. Also please note that there is no sequential order between every method and the next one — except for the preprocessing step and topic modeling one. In other words, these steps can be performed in parallel and the order you see in the pipeline is just how we chose to run the multi-dimensional analysis. Next, we check how hateful the data was using a pre-trained hate speech classifier model by Davidson et al. [16]. The reason why we didn't design our own classifier is because in order to do so we need a sufficient amount of labeled data but our data is large, noisy and unlabelled. Training a classifier using such data means labeling it by human annotators which challenges the goal of decreasing human exposure to raw data even if it was only for labeling and annotation purposes.

That is why we opted instead to use one of the existing good hate speech classifiers, it was trained using online data — that is believed to contain hate speech — and achieves a good accuracy. So by the end of this step, we obtain a class label assigned to each document and these labels were (Hate Speech, Offensive Language, Neither). Another motivation behind choosing this particular classifier and its classes is because unlike binary classes (Hate or No Hate) this classifier differentiates between what is fully hateful text, what can be considered offensive (a genuinely lesser degree of toxicity) and what is neither. An example of both classes would be:

- classified as Hate Speech: "@JuanYeez shut yo beaner ass up sp*c and hop your f*ggot ass back across the border little n*gga and RT @eBeZa: Stupid f*cking n*gger LeBron. You flipping jungle bunny monkey f*ggot."[16].
- classified as Offensive Language: "When you realize how curiosity is a b*tch"[16].

Then, we leverage a seed keywords list related to hate, abuse and kinetic actions - which along with the help of domain experts was created and approved. An example keyword would be an action like rape, murder, or slaughter. Our *MFA* performs query expansion on this list to increase its size and make it more comprehensive then we parse all documents to look for any occurrence of these keywords and identify the records with any possible actions and targets. We do so using the NLP task Part of Speech Tagging. A hypothetical example would be: *" ... I wanna rape those mo***er f**kers animals!! cant you see ... "*, here the keyword "rape" is occurring as a "Verb" and "animals" is occurring as an "Object" and is considered the target in this. After that step, is the identification of the involved entities, in which we are interested in extracting about whom the speaker is talking, in other words, which country, person, time, organization, religion and so on. We do so using the NLP task Named Entity Recognition. A hypothetical example would be: *"... it would have been better if the People in the European Union act to ... "*, here "People", "European" are extracted as "PERSON" and "GEO" entities respectively.

A key element and the significant part of our approach is in its final step, in which we make use of all the previous results by coming up with a modifiable and simple scoring scheme that accumulates all the different steps outcomes for all the records (documents) such that each step has its own weight that contributes to the overall score for that document. After this step, all the documents are assigned a calculated overall score that gives an indication of its priority. After that, we then use this embedded score component to design an information retrieval application that returns a ranked list of the data such that documents with the highest score are placed on top and least ones are below. The illustration in figure 3.1 below gives a visual representation of the overall process.

3.3 What is our Data ?

In this proof of concept study, we are using data that was provided to us for research purposes by the Canadian Association for Security and Intelligence Studies Vancouver (CASIS). CASIS is a nonpartisan, voluntary organization established in 1985. Its purpose is to provide informed debate in Canada on security and intelligence issues. Data was originally scraped by Hope Not Hate and provided to CASIS and the Canadian Centre for Identity-Based Conflict by The Canadian Anti-Hate Network. Also, it is worth mentioning that the source of this data is the dark web and because of a data disclosure agreement with data providers we can not provide original examples or state the source for confidentiality reasons.
This data is particularly interesting for two main reasons:

- Firstly, it consists of a large amount of unlabelled, unstructured and unprocessed data (raw data).
- The second reason is related to the subject matter of the data content. This data contains disturbing, abusive and hate speech content that can possibly lead to some harmful psychological side effects on the reader — even if it is a domain expert — especially when it involves reading or analyzing a huge amount of it, which is one of the main motivations for this study in the first place.

Regarding the data size used in this work, we only worked on a sample set of 11 thousand records out of the whole data (which is around 1 million records) in which each record contains fields such as (user id, content posted by user, category, replyID). An essential part of the study was the effort directed towards structuring and preprocessing the data in a way that would enable us to be able to apply the different natural language and machine learning methods on it.

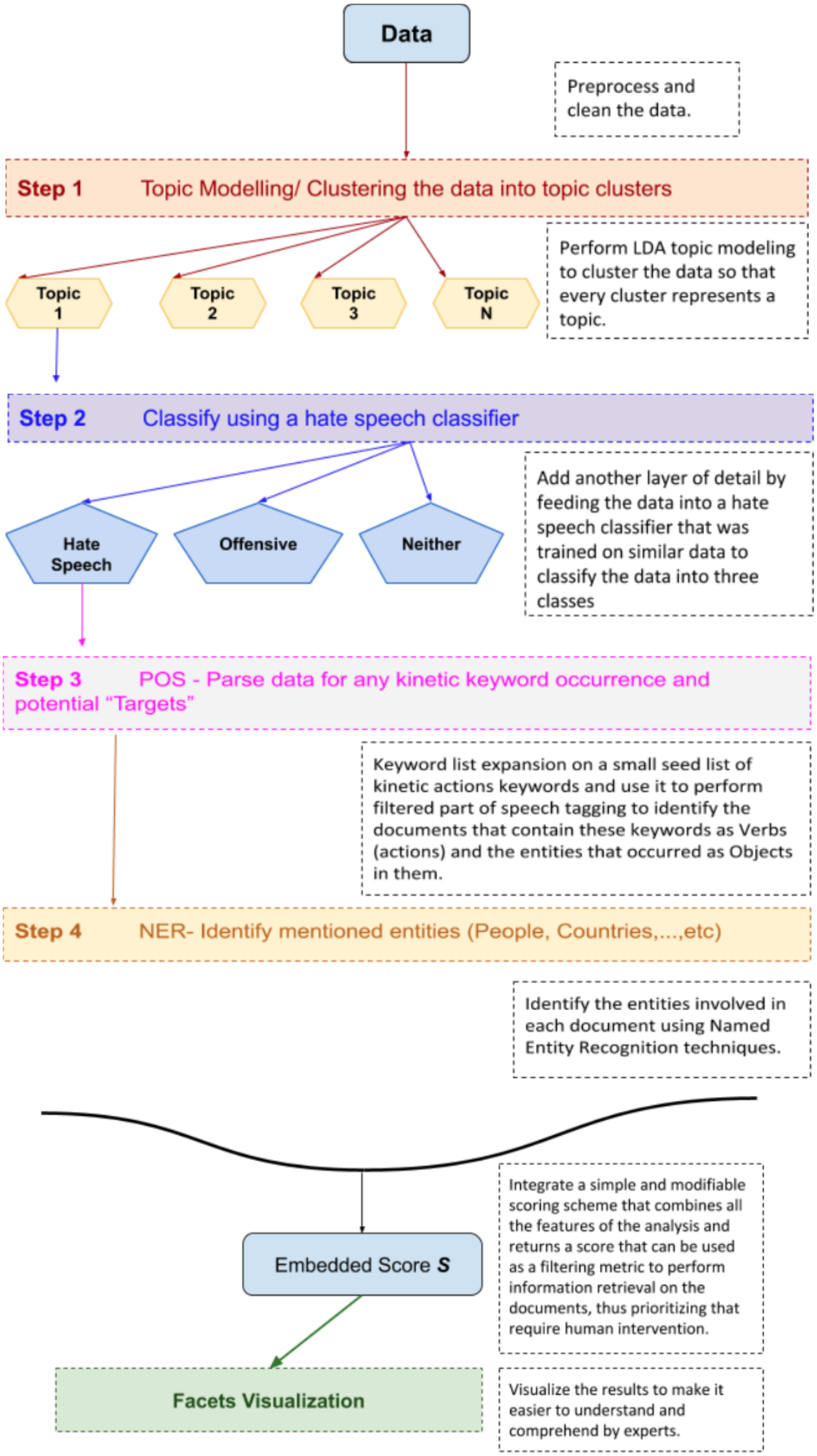

Data
Preprocess and clean the data.
Step 1 Topic Modelling/ Clustering the data into topic clusters
Topic 1
Topic 2
Topic 3
Topic N
Perform LDA topic modeling to cluster the data so that every cluster represents a topic.
Step 2 Classify using a hate speech classifier
Hate Speech
Offensive
Neither
Add another layer of detail by feeding the data into a hate speech classifier that was trained on similar data to classify the data into three classes
Step 3 POS - Parse data for any kinetic keyword occurrence and potential "Targets"
Keyword list expansion on a small seed list of kinetic actions keywords and use it to perform filtered part of speech tagging to identify the documents that contain these keywords as Verbs (actions) and the entities that occurred as Objects in them.
Step 4 NER- Identify mentioned entities (People, Countries,...,etc)
Identify the entities involved in each document using Named Entity Recognition techniques.
Embedded Score S
Integrate a simple and modifiable scoring scheme that combines all the features of the analysis and returns a score that can be used as a filtering metric to perform information retrieval on the documents, thus prioritizing that require human intervention.
Facets Visualization
Visualize the results to make it easier to understand and comprehend by experts.

Figure 3.1: Visual representation of the overall process of *MFA*.

Chapter 4

System Implementation

This chapter goes over the different processing stages that were introduced in chapter 3. It details every step with respect to each step's implementation.

First, we explain the data preprocessing that we performed on our raw data to transfer it from human-readable into machine-readable. Then, we talk about the clustering technique and how we used topic modeling analysis to cluster the data. This corresponds to step 1 in the previous chapter. Next, we talk about the labeling step mainly about hate speech classification, introduced in step 2. We explain in detail the existing model that we used and how it incorporated in it different components such as sentiment analysis. After that, step 3 in our pipeline, we move to the linguistic analysis in which we collaborated with domain experts to build a seed keywords list that is related to hate speech and kinetic actions — list of action words that are more likely to indicate an escalation from online hate to real-world crimes. We discuss the methodology we used to build it and how we performed query expansion on this list to make use of it. We explain how we made use of the expanded list to identify possible actions and targets. We illustrate the technique called Part of Speech Tagging POS which we used to identify the records that contain kinetic action keywords that were used as verbs in other words implying an action. Then, in step 4 in our pipeline, we discuss the involved entities identification step and the NLP method we used for it Named Entity Recognition - NER. Finally, we detail the proposed scoring scheme that will incorporates all the previous steps. This corresponds to the scoring scheme step.

4.1 Data Preprocessing

In natural language processing, the process of cleaning and preparing the data is often referred to as data preprocessing. Real-world data is usually incomplete, inconsistent, maybe lacking in certain aspects, and is likely to contain many errors. The original data that was provided to us was in its raw format, for example, contains many misspellings, errors,..., etc. Our goal in this step is transforming such unstructured and unclean textual data into a format that can be used and is suitable for further application of natural language and machine learning methods.

Preprocessing is often done in multiple steps as follows: first is normalization which works on converting letters to either lower or upper case depending on the use case, converting numbers into words, removing white spaces, punctuation and abbreviations expansion. Another step is stopwords removal, stopwords are referred to as the words that are most common in the English language. Stop words are for example (the, is, at) and they do not carry any significant meaning in a sentence so they are removed. Next, is a process called tokenization, in which a character sequence from a document unit is chopped into pieces, called tokens, perhaps at the same time throwing away certain characters, such as punctuation. For example, the sentence *"Trees are full of leaves"* would be broken down into (Trees, are, full, leaves).

Next is a process called stemming. It refers to the process of reducing a word to its root form, for example, drive, driven, driving, drove all have the same root. Similar to stemming is another NLP task called lemmatization, which is basically converting a word to its dictionary format making use of lexical knowledge bases to get the right base form of the word just as in a dictionary. The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. However, the two terms differ, stemming usually refers to a process that chops off the ends of words in the hope of achieving this goal correctly most of the time, and often includes the removal of derivational affixes. Whereas lemmatization usually refers to doing things with the use of vocabulary and morphological analysis of words, normally aiming to remove inflectional endings only and to return the base or dictionary form of a word, which is known as the lemma. We will provide more details in chapter 4.2 on how we chose to perform the lemmatization for the clustering step (LDA topic modeling training).

4.2 Clustering

After the preprocessing step is done, next on our MFA pipeline is clustering. Document Clustering is the process of placing documents into different clusters such that the documents that are placed in the same cluster are more similar and relevant to one another compared to the rest of the documents in the text corpus. There are two types of document clustering algorithms, hard clustering, and soft clustering. The first as from its name performs hard assignment meaning it assigns each document to exactly one cluster (group). The latter performs soft clustering meaning that a document can belong to more than a single cluster at the same time but in different distributions for each cluster. In other words, a document might be 70 percent in cluster 1 and 30 percent in cluster 2. An illustration of these two is shown in figure 4.1 below:

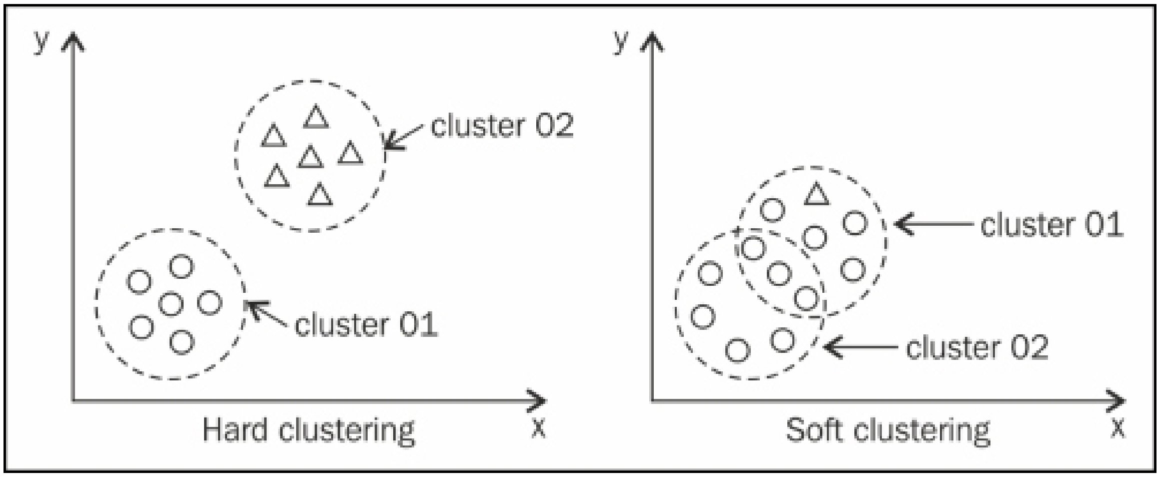


Figure 4.1: Hard and Soft Clustering. Source: [65].

Moving back to our pipeline, since our source data was unstructured we thought of different ways to convert it from unstructured to structured data. A meaningful way that would provide more insights and discover new things about our data and would help us achieve our objective of providing a framework to help explore inflammatory unstructured data more efficiently and healthily. Ultimately, to help redact data exposure to domain users. Clustering could have been performed using different features such for example by splitting the documents based on the UserID field or by using the category field that gives an abstract idea about the document content. However, for us, since we wanted a way that would provide deeper and learned knowledge about our data the best idea was to perform clustering based on topics discussed in the data. That is why we performed topic modeling for clustering.

As we explained in chapter 2, the description of a topic model is a statistical and probabilistic model for discovering the abstract topics that a collection of documents talks about. Topic modeling is a frequently used text-mining tool in information retrieval and data mining for discovery of hidden semantic structures in a corpus. The options we had for clustering via topic modeling were to use one of the unsupervised learning clustering algorithms — since our data is unlabelled. Such algorithms include K-means Clustering, Fuzzy Clustering or Latent Dirichlet Allocation (LDA). We will provide some brief explanation of our two best suitable options (k-means and LDA) and illustrate why we eventually decided to go with LDA as our clustering algorithm.

The definition of *K-Means Clustering Algorithm* in data mining is an unsupervised learning algorithm that clusters the data (textual documents) into K disjoint clusters. It works well with the documents that are clear-cut and well-separated. It is also considered to be one of the simplest clustering algorithms and it works by specifying k centers such that each represents a cluster and then assign data points to the nearest center (cluster) and keep re-calculating these center points until the in-cluster sum of squares reaches a local minimum (since this problem is NP-hard). The goal is to keep the centroids as small as possible. An example is illustrated in figure 4.2.

In order to perform K-means we would follow the authors proposed k-means algorithm which is described in [22] and [34]. There are existing good implementations for it using Python and different libraries like Scikit-learn. However, we didn't attempt and perform k-means since we figured LDA would work better in this type of study as it does soft clustering (a document might belong to more than one cluster and we pick the topic the cluster that it represents the most with the highest distribution) unlike K-means which does hard clustering (hard assignment i.e. a document can belong to only topic cluster). In general, both, k-means and LDA works fine with the task of clustering via unsupervised learning.

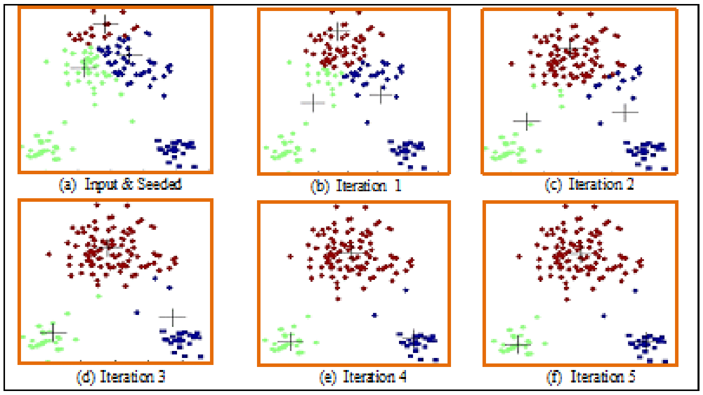


Figure 4.2: K-Means Example as explained in [26].

Our other alternative for clustering was using Latent Dirichlet Allocation (LDA). As we previously explained in **section 2.2.1** *the LDA Algorithm* is an unsupervised learning algorithm for text clustering and topic modeling. LDA assumes that every document contains a set of topics rather than a single topic and each topic is represented by a set of keywords out of all the words in these documents — a corpus of text. LDA's main job is to identify these keywords in a way such that each topic is like a cluster of keywords along with the probability of how much each keyword is occurring in that topic cluster. Also, the same keyword may occur in multiple documents but it would have a different distribution per document. Mostly, this set of keywords per topic helps us with interpreting what is the topic and in making sense of what is the discussed topic. We have already explained in detail how this algorithm works and its significance so we will not repeat it. However, it is worth mentioning that both K-Means and LDA would in principle work well for this task but the main difference for us was that the K-Means algorithm did Hard Clustering whereas the LDA algorithm did Soft Clustering. Thus, we chose LDA as it gave us a more realistic and reflective result for our analysis.

Now that we have explained all the concepts needed and background context, we will explain the details of our implementation of LDA. There are multiple libraries for LDA topic modeling, most famous are the ones in Gensim [1] and Sci-Kit Learn [2] libraries. Comparing the two, we can not say that one is evidently better performer than the other since there is no single metric of performance to compare against. That is why, we simply chose to build and train an LDA model [3] using the Gensim library because it provides more built-in functions related to topic modeling.

As part of preprocessing for topic modeling in LDA model training we did lemmatize our training data, however, we kept only these part of speech words (allowed-postags=['NOUN', 'ADJ', 'VERB', 'ADV']) as these are the ones which affect the most the meaning of a sentence and a topic in general. There are also different tuning parameters that we can use to train and fine tune the model but the most important ones are:

1. An id2word mapping matrix of our data — it is basically a dictionary that maps a unique id for each word.

2. Our data corpus that captures the Term-Document frequency of our documents. We take the preprocessed and cleaned data from the first step in section 4.1 and consider it as our training corpus. Then, we compute the Term-Document Frequency for the corpus using Bag-of-Words and the id2word matrix that we created in previous step. We convert our corpus into a matrix of bag-of-words[4] vectors. For example an entry of (11, 1) in the matrix would mean that word id 11 appears once in the document.

3. A priori number of topics k that we arbitrary choose (we will explain shortly how we decided it is the right number of topics for our data and model).

4. α, and β, are hyperparameters that affect sparsity of the topics and we explained these two parameters in 2.2.1. According to Gensim, both default to $(1.0/k)$. As we explained before α, corresponds to the Document-Topic weight whereas β, corresponds to the Topic-Word weight. Tuning these two hyperparameters deals with the topic distribution with respect to both to documents and words. We tried training with different values of both and the best performance was yielded by the defaults.

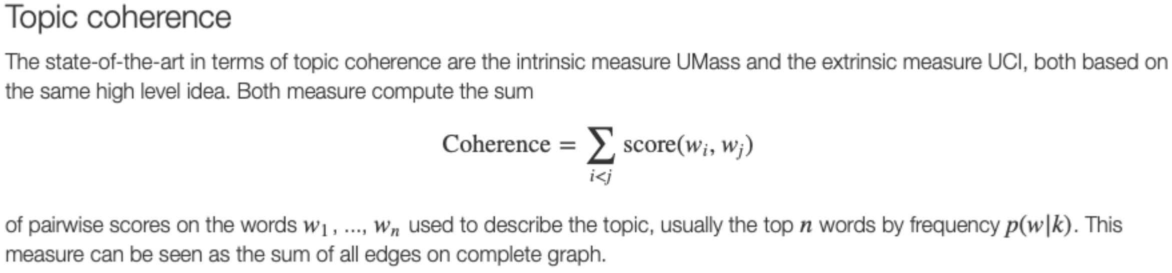
[1] Wiki: Gensim is an open-source library for unsupervised topic modeling and natural language processing, using modern statistical machine learning. https://pypi.org/project/gensim/, https://radimrehurek.com/gensim/

[2] https://scikit-learn.org/stable/

[3] We inspired our implementation from https://www.machinelearningplus.com/nlp/topic-modeling-gensim-python/

[4] https://en.wikipedia.org/wiki/Bag-of-words_model

After we have prepared all the needed parameters, we train an LDA model using them and with k number of topics set to 20, which after testing appeared to be the optimal number of topics fitted for our data. We tested other possible values of k and 20 gave the highest score of coherence (0.4216). A high coherence score means that the topics which were generated by the LDA model are more human-interpretable. Human-interpretability is the metric that we use to evaluate how good or bad the performance of a topic model is. Usually, a bad LDA model would have a low coherence score and its generated topics (the top words in each topic cluster) would be hard to interpret for a topic by a human reader. For reference, we used the coherence c-v Measure function that is supported by Gensim. Figure 4.3 explains the main idea of the topic of coherence calculation. Please refer to [50] work on "Exploring the space of topic coherence measures" for more explanation on coherence in topic modeling.



Topic coherence

The state-of-the-art in terms of topic coherence are the intrinsic measure UMass and the extrinsic measure UCI, both based on the same high level idea. Both measure compute the sum

$$\text{Coherence} = \sum_{i<j} \text{score}(w_i, w_j)$$

of pairwise scores on the words w_1, ..., w_n used to describe the topic, usually the top n words by frequency $p(w|k)$. This measure can be seen as the sum of all edges on complete graph.

Figure 4.3: Topic Coherence Illustration by Quentin Pleple. Source: [48].

Now that we know how we found our optimal number of topics, we will talk about the topics we found and how each keyword in a certain topic cluster has a distribution value that indicate the weight of this keyword in this topic cluster. Figure 4.4 is a 2D visualization of our results where each circle is a topic cluster and the size of the circle indicates how many documents belong to this cluster i.e a large-sized cluster circle means that this topic occurs in many documents in our data and a small-sized one indicates the opposite. Since we were able to find a good number of clusters (k = 20) all of the topic clusters have a fair similar size to one another which also shows that our clustering was done quite well.

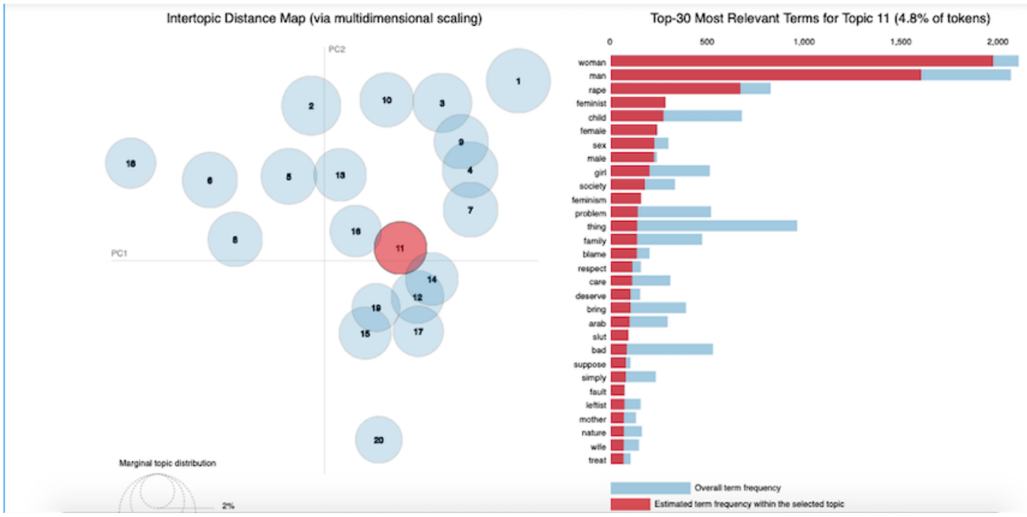


Figure 4.4: Example of LDA topic clusters along with keywords probabilities. Blue corresponds to overall frequency and red corresponds to estimated term frequency within the selected topic. Visualization is done via pyLDAVis which is a commonly used library for topic modeling results visualization.

Another thing we did to visualize the clusters of our documents in a 2D space use the T-SNE (t-distributed stochastic neighbor embedding) algorithm, which is a machine learning algorithm that was created by Maaten et. al in 2008 [36]. It is used for visualization of high dimensional data and it is commonly used due its significance as a visualization method in documents topic modeling. This visualization tells us that we have 20 topic clusters with each document presenting a dot in the graph. So the density and width of each cluster represent how many data-points it has in it, a big and dense one indicates that this topic is discussed more with respect to the rest of the topics. The distance between each data-point and the centroids of clusters indicates the distribution of how much this document is in this topic cluster. In other words, a document's data-point placed exactly between two clusters means that it discusses these two topics equally — the same idea of the soft clustering we discussed before. Also, one can draw from the visualization and each cluster size that we have a relatively good split among the documents. In other words, there is no one or two clusters that dominate the whole documents. Figure 4.5 below shows the overall structure of our LDA model results.

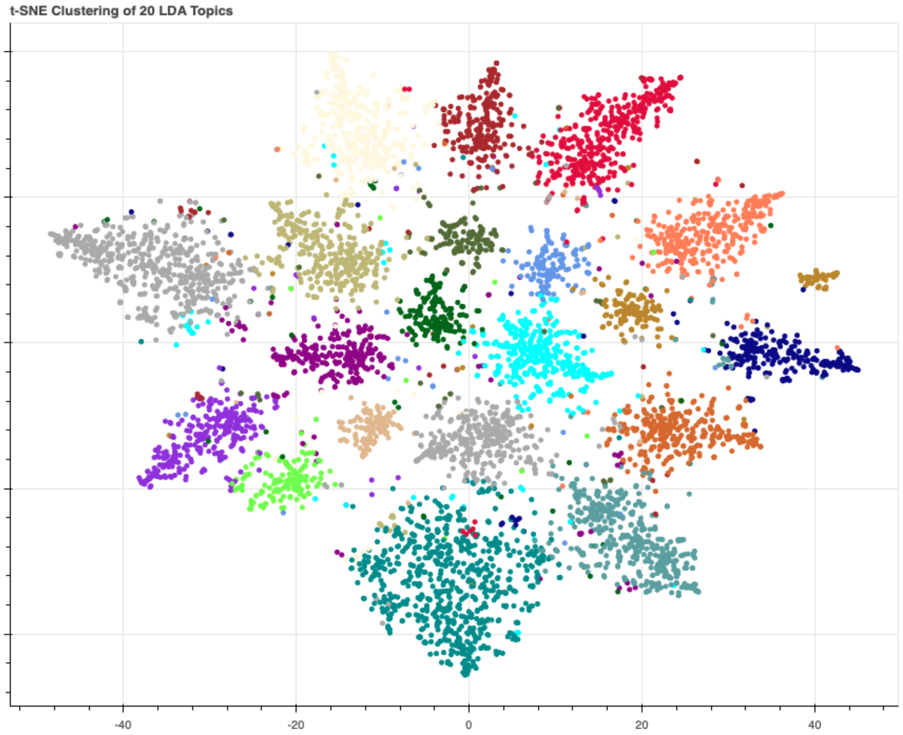


Figure 4.5: T-SNE Visulaization of LDA Topics

To summarize, in this step in our pipeline we simply used topic modeling to transform our data from unstructured to structured via clustering our documents. We explained the choices we made such as which unsupervised learning algorithm for clustering to use (K-Means or LDA) and the number of topics (clusters), k, should we pick. We did this using LDA topic modeling and we implemented it via Gensim and NLTK libraries. Also, we showed some of the visualizations we did for this step as the remaining ones are more revealing and descriptive of the data which is to remain private in this study analysis.

4.3 Labelling

When we first started with this analysis and study on the data given to us the first thought was to design a regular hate speech and toxicity classifier and detection model since we were interested in early detection and elimination of such content and also to prevent any escalation from online hate speech to real-world offenses and crimes. However, we were a little bit ambitious because after thorough research and observations on our data we realized that this is simply not feasible given the current human resources, the reasons that led us to this conclusion can be summarized as follows:

1. Our data is enormous in size, which is usually a good thing except when this data is unlabeled and unstructured. Labeling such data means letting human annotators manually go through at least a sufficient amount of it to be able to have a training data set in case we were to design an automated detector model using machine learning and NLP. Of course, since the main objective for us was to redact and reduce human exposure to offensive and psychologically disturbing content this approach of manual annotation was dismissed.

2. We lacked sufficient amount of real-time examples of records of online hate speech moving to become real-world crimes and offensives, in machine learning and NLP language we lacked the True Positives (TP)[5] in our dataset. Although, we had a few of them, it simply wasn't enough to build a classifier using them only such that it would successfully distinguish whether a certain document, comment or a post by a specific user on a certain topic would result into a hateful and kinetical action against someone or something in the offline world or not.

Regardless of the mentioned challenges, we still wanted to be able to know whether a document (record) is hateful or not using different factors such as the sentiment of the language used in the text. So given the limitations we had, we opted instead to use an existing Hate Speech and Offensive Language Detection and Classification model that was trained on social media data that is of somehow similar content to our data. In 2017, Davidson et al. designed a multi-classifier model that would classify a given textual input into three classes [Hate Speech, Offensive Language, Neither]. Their work titled "Automated Hate Speech Detection and the Problem of Offensive Language" [16] used a crowd-sourced hate speech lexicon to collect tweets that contain certain hateful keywords and they also used crowd-sourcing to classify a training dataset sample into the three mentioned classes, then, they use this dataset to train a classifier to distinguish between these three categories. And in regard to their model details, the authors tried with different types of models and

[5] A true positive test result is one that detects the condition when the condition is present. Source: `https://groups.bme.gatech.edu/groups/biml/resources/useful_documents/Test_Statistics.pdf`

ended up using a logistic regression with L2 regularization as the best performer (F1 score of 0.90) and modeling was done using Scikit-learn. We chose to use their model for two main reasons. First it is in the same domain that we are interested in and the nature of the data they used in their analysis was quite similar in nature to ours but less harmful — since the data was collected from a known social media website "Twitter" and the content posted there is monitored to a certain degree by platform rules. Second, it is different from other previous work done in this area in the sense it does not rely only on lexical methods to detect hate speech and offensive language. Rather it embeds other features in their model building like for example the sentiment feature. By lexical methods we mean whether a document contains a certain keyword or not.

So, we used their pre-trained model as a classifier for ours, we fed our data and got a class for each record (document) and added this as a new column to the structured data from the previous steps. Also, regarding the accuracy of the classifier performance on our data, in particular, we randomly sampled 300 documents (100 per each class, in other words 100 from each the classified as "Hate Speech", "Offensive Language" or "Neither") of the classifier results on our data and had human experts classify them but without letting them know what the model class was and we then used the experts classification as our ground truth. After that, we did some accuracy measures and we calculated the Recall and Precision and they came out sufficiently accurate for us to adopt it as our classifier. The result of our confusion matrix on the 300 sample records are shown in figure 4.6.

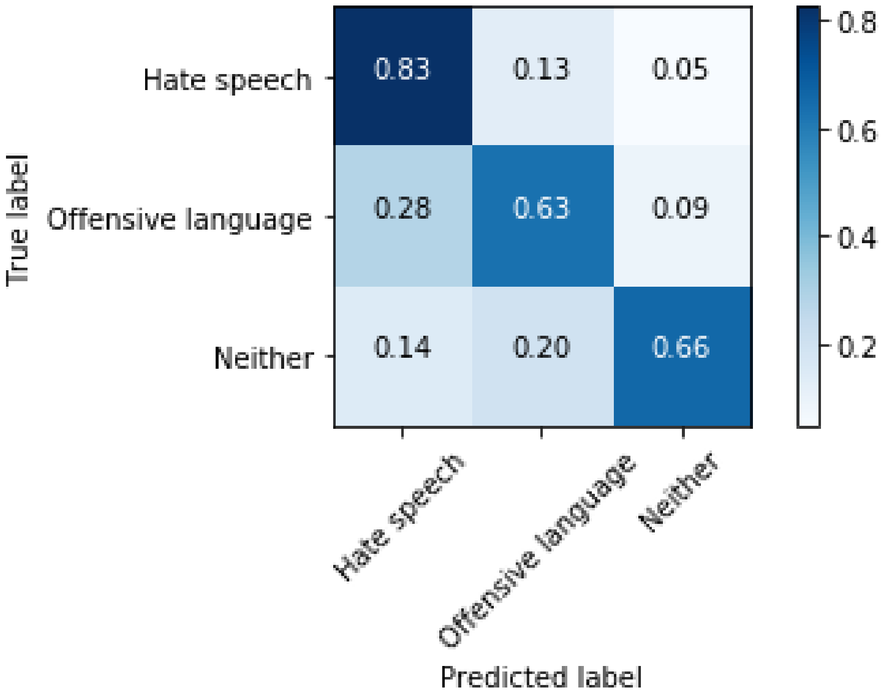


Figure 4.6: The result of our confusion matrix on the 300 sample records.

To summarize, in this step we used an existing pre-trained Hate Speech and Offensive Language Classifier on our data and obtained a class for all of our records. We explained what are the challenges that we faced and why we decided to use this particular model. We also show the confusion matrix results we obtained on the 300 randomly sampled records.

4.4 Seed Keywords List Creation

Another angle of our multidimensional analysis was using the most common technique in such tasks which is to create our lexical resource from a small seed keywords list that is related to the domain and use this seed list to design an Information Retrieval model. This is the third step of our *MFA* approach. In section 4.1.1 we will explain how we created this seed list and in section 4.4.2 we will go over how we performed query expansion on it to expand and enrich our list.

4.4.1 Lexical Resources

Since it is more likely the case that abusive and hate speech documents would contain keywords of a certain type, such words mostly would be sentimentally negative like for example abusive swears or insulting sarcasm. In our case, we were interested in the domain of hateful documents that would encourage or incite or motivate an escalation from online hate to offline hate actions. Since this seed list would be the foundation for all of the expanded keywords, it was critical to include the effective keywords for us to have a thorough expressive list. So, to obtain the right set of words — alarming and predictive words — we had to look for an appropriate lexical resource and for us, this resource was via collaborating with domain experts to help craft and come up with a seed list of words of interest.

We simply asked a straight forward question to the subject domain experts as follows: *what are the words that if they did a quick skim read of the document and found them, the document would catch their attention and it would increase the overall importance of this particular document to require a further deeper analysis from them ?*

The final seed list was quite small in size, around 20 words and an example of such kinetic action keywords would be: [Rape, Burn, Death, Fight]. Existence of any of these words in a document would increase the importance of it.

4.4.2 List Expansion

After we have created a seed list of keywords that are considered to have a high weight of importance according to subject experts and analysts, we performed an expansion on it to increase its size and widen it by adding more words that are semantically or contextually similar. We did it via ia Thesaurus resources, such as WordNet and Dictionaries Synonyms: we used NLTK WordNet library to look for synonyms of words in our list. Wordnet is a lexical database for English that was created by Christiane Fellbaum in 1998 [20]. WordNet is seen as a combination of dictionary and thesaurus. It groups English words into sets of synonyms, provides short definitions, examples, and relations among these synonym sets.

The results of this expansion would be for example for the keyword "Kill", results in adding the following keywords to our list: ["killing", "toss off", "defeat", "stamp out", "kill", "drink down", "obliterate", "shoot down", "pour down", "putting to death", "vote out", "wipe out", "belt down", "bolt down", "vote down", "down"].

After applying the mentioned technique we are able to successfully widen and expand our seed list which we will make use of in the following step.

4.5 Actions and Targets Identification

Now that we have a relatively representative and wide-ranging kinetic actions keywords list, we utilize this resource to help us identify the documents that contain disturbing and alerting content to be able to mark them as higher value in the context of a toxicity and severity metric. We refer to a well-known and common technique in NLP called Part-of-Speech (POS) Tagging, that was already explained in section 2.4.1 but it is simply the process of assigning labels for words in a sentence such that it takes into account both the context of each word and the lexical category of it. These labels are called POS tags and an example of such tags are for example the labels (Verb, Adjective, Noun). In this step, firstly, we parse the documents looking for any occurrences of any of the kinetic keywords in them and then, we apply POS tagging to look into these occurrences for the ones that happen to be used as "VERBs" only — since we are interested in studying when an online hate speech escalates into real-world kinetic actions as a "VERBs" as the goal is to early identify disturbing or alerting escalation between the two worlds.

An extra zoom in for these same records that contained verbs from this list we use POS tagging to extract the entities with the tag "OBJECT". These entities might be possible potential Targets in the same sentence in which the kinetic action keyword occurred as a Verb. A theoretical example would be if we have the sentence:
"*... rape all these **** retarded people ...*", the extracted information would be "rape" as a "VERB" here would be identified as one of the kinetic keywords in the list and it occurred as a verb and "people" as an "OBJECT" would be the potential target with an "OBJECT" tag.

To summarize, by the end of this step we have analyzed the documents using the linguistic features of the text in each of it using POS tagging. We identify the documents containing any of the kinetic action keywords and extract these keywords along with any found objects.

4.6 Involved Entities Identification

The next step in our *MFA* approach is the task of identifying the involved entities in each document. Sometimes, a document weight value is hugely affected by which entities it is talking about especially as in our case where documents contain inflammatory content that may contain possible real threats to the entities mentioned. Also, after discussions with domain analysts, they mentioned how sometimes they would be looking for certain entities in a document such as a certain country or religion or person. So it is rather an important task for an approach as ours to identify those entities.

That is why in this step, we are interested in identifying the entities involved in the documents. To do so, we execute a process called Named Entities Recognition NER, which we already explained in detail in section 2.4.2. It is defined as the process of identifying the named entities in a document, paragraph, sentence or the whole of a text corpus. Those entities could be the persons mentioned, organization, location or geopolitical entities. It simply assigns a tag that indicates what type of entities this word (token) — sometimes more than one word — represents.

After trying different existing Python libraries implementations of NER such as spaCy and Stanford-NER we found the highest performance accuracy on our data was yielded via the spaCy model, so we adopted it. As for some background on these two libraries, Standford-NER is a java implementation for NER recognizer that was developed by Finkel et al. in 2005 [21]. It uses a conditional random field classifier to design a multi-class language classifier. spaCy is open-source, written in Python and Cython and its original author is Matthew Honnibal [30]. It can recognize various types of named entities like a person, or a country in a document, by asking the trained model for a prediction. The predictive model can be retrained and updated with new annotation examples of what constitutes an entity or if there are any newly emerged ones by simply retraining the existing model with an updated dataset, in other words, a certain group or political party name [6].

A theoretical example of the results would look like:

$$(\text{“}Obama\text{”}, \text{“}PERSON\text{”}), \tag{4.1}$$

$$(\text{“}France\text{”}, \text{“}GPE\text{”}), \tag{4.2}$$

$$(\text{“}TheClairssFoundation\text{”}, \text{“}ORG\text{”}), \tag{4.3}$$

$$(\text{“}Jews\text{”}, \text{“}NORP\text{”}). \tag{4.4}$$

As a reference [7], the labels here stand for:

- PERSON stands for people, including fictional.
- GPE stands for geopolitical entity, like countries, cities or states.
- ORG stands for companies, agencies, institutions.
- NORP stands for nationalities or religious or political groups.

To summarize, in this step we have identified and extracted the involved entities for all documents (records) through NER using spaCy, which gives us another aspect and dimension of information about the data.

4.7 Confidence Scoring Scheme

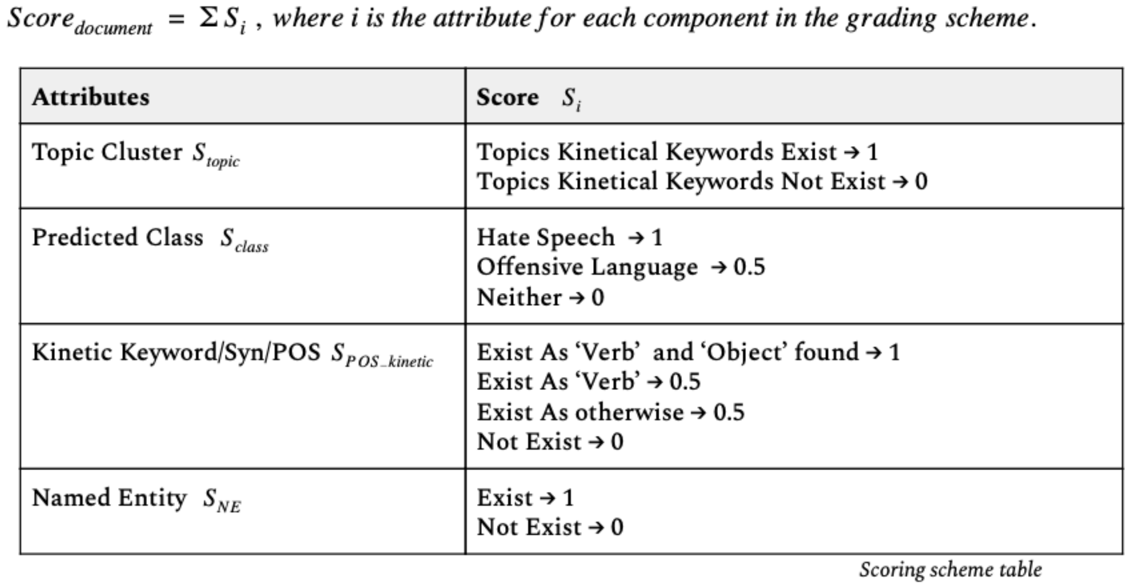
Now that we have ran different and multiple NLP and machine learning techniques on our documents, we realized that we want a way to help us understand how all the previous processes contribute to the value of a document. A simple yet meaningful way, something that can work as an indicator for analysts. That's where the cumulative confidence score idea came in place. The final step of our *MFA* approach is the one that combines together all the previous results and makes sense of them using a ranking score per document based on what this document's outputs were in every step before.

The value is being able to comprehend how a document is scored based on a sum of human-engineered features, linguistic features, and computational features. This confidence

[6] Website: `https://spacy.io`

[7] The full notations doc is here `https://spacy.io/api/annotation`

score will work as both an indicator of this document's content richness and as a mean to perform information retrieval of the documents such that the documents with the highest score should be placed at the top to be accessed first by the human analysts. This serves both the objectives of extra time efficiency and redaction of inflammatory content exposure. To create such a score, we defined a scoring function with certain criteria to use them for score calculations.



$Score_{document} = \Sigma S_i$, *where i is the attribute for each component in the grading scheme.*

Attributes	Score S_i
Topic Cluster S_{topic}	Topics Kinetical Keywords Exist → 1 Topics Kinetical Keywords Not Exist → 0
Predicted Class S_{class}	Hate Speech → 1 Offensive Language → 0.5 Neither → 0
Kinetic Keyword/Syn/POS $S_{POS\_kinetic}$	Exist As 'Verb' and 'Object' found → 1 Exist As 'Verb' → 0.5 Exist As otherwise → 0.5 Not Exist → 0
Named Entity S_{NE}	Exist → 1 Not Exist → 0

Scoring scheme table

Figure 4.7: Confidence Scoring Scheme Table.

The scoring function is really simple yet highly effective and can be used as a base foundation that can be built upon and modified with additional components. You may also notice that we kept the scoring criteria as simple as possible so that to make it easier for any future built-upon modifications. We have four components in it as follows:

- LDA Topic Cluster Kinetic Keyword Score: this component score is based on section 4.2. After the clustering, we gathered all of the top keywords from all the topics (clusters) and with the help of domain experts we identified a subset out of it with the keywords that are more prone to kinetical actions, hate speech and toxicity in general. So, in this component, if the document is placed in a certain topic cluster that contains any of the keywords from our subset then the component score is 1, otherwise, the score is 0.

- Predicted Class Score: this component score is based on step 4.3. If the document content was classified as "Hate speech", then score is 1, else if it was classified as "Offensive Language" then score is 0.5, otherwise score is 0 — class "Neither".

- Kinetic Keyword/Synonym Score: this component score is based on both steps 4.4 and 4.5 combined. If the document contains a kinetic keyword from the expanded list and it is linguistically occurring as a "Verb" along with an identified "Object" in the sentence then this component score is 1. Otherwise, if it just occurring then component score is 0.5 and if it is occurring as "Verb" then add 0.5 on it to have a total score of 1 for this component in that case, otherwise, it is 0.

- Named Entity Identified Score: this component score is based on step 4.6. If the document contains any involved entities then its score is 1, otherwise, it is 0.

To summarize, in the final step of our *MFA*, we used the criteria defined in the table in 4.7 to implement a scoring function that incorporates all previous steps. We calculated the embedded score for all documents based on their results in previous processes.

In the next chapter, we illustrate how we made use of this score to visualize and perform information retrieval.

Chapter 5

Visualization

In this chapter, we describe the concept of multifaceted representation, information retrieval and navigation of data in section 5.1, and the Flask Web app that we have built to demonstrate the idea we are proposing in our *MFA* in section 5.2.

5.1 Multifaceted Representation and Information Retrieval

In developing our *MFA* we wanted both a representation of our analysis that would serve the objective of redaction of exposure to hurtful content but yet at the same time would not affect the overall efficiency of human analysts' activities or limit their knowledge retrieval. After some thorough research in the areas of document visualization and information retrieval which are described in the next subsection, we came across the concept of Faceted Navigation that turned out to be the best. In section 5.1.1, we will provide some background knowledge about this approach and in section 5.1.2 we will describe the concept we adopted for our *MFA* approach with some sketches to illustrate the concept.

5.1.1 Background

Faceted Navigation in the words of Marti Hearst in her book "Search User Interfaces" [27], which was published by Cambridge University Press in 2009 is: "a solution to the problem of strict navigation which results by assigning documents to single categories when in fact most documents discuss multiple topics simultaneously". Hearst defines faceted navigation as an approach that describes documents by a set of categories, as well as attributes (such as source, date, genre, and author), and provides good interfaces for manipulating these labels by building a set of category hierarchies such that each corresponds to a facet (dimension or feature type) that is relevant to the collection to be navigated — each facet has its own hierarchy of terms [27]. In an earlier work, Hearst et al., 2002 [28], faceted metadata was defined as being made of orthogonal sets of categories, such that each category corresponds to a different and independent aspect of the document.

The concept of faceted navigation sometimes also called as faceted search, guided navigation or even parametric search. Most of the time these terms refers to the same idea of assigning multiple labels and categories to items (or documents) such that it would enable dynamic clustering among these categories in each facet. Figure 5.1 below illustrates the concept of faceted metadata, faceted navigation or facets search.

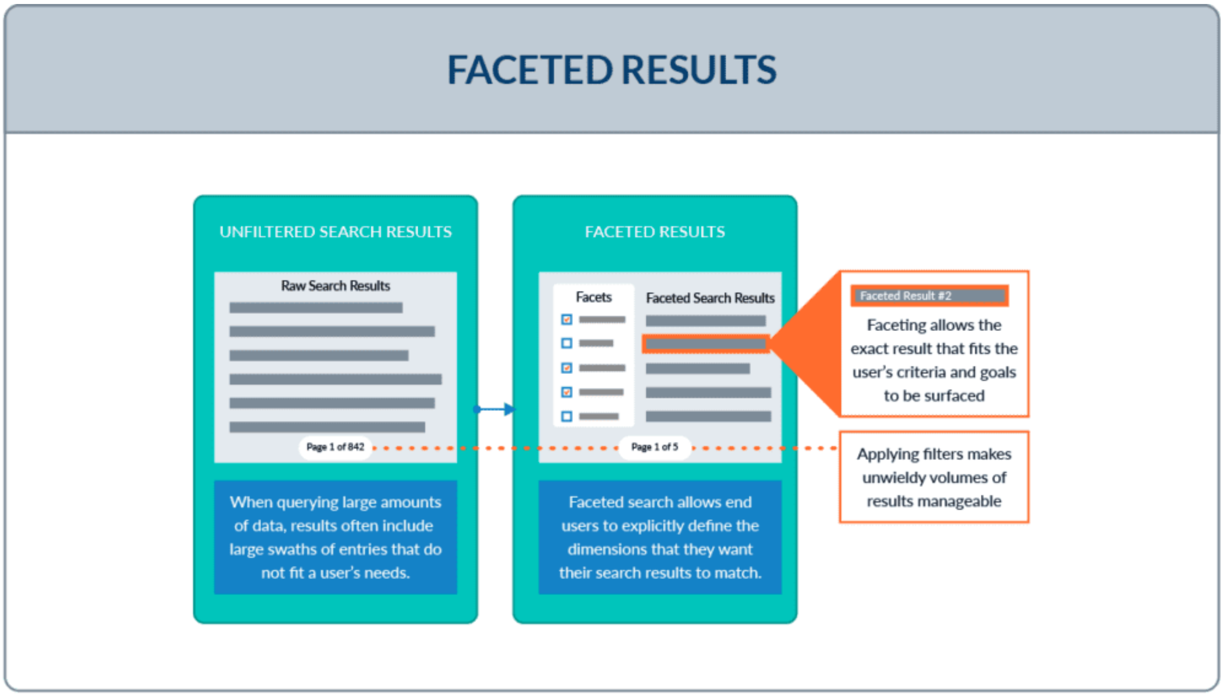

FACETED RESULTS
UNFILTERED SEARCH RESULTS
Raw Search Results
Page 1 of 842
When querying large amounts of data, results often include large swaths of entries that do not fit a user's needs.
FACETED RESULTS
Facets
Faceted Search Results
Page 1 of 5
Faceted search allows end users to explicitly define the dimensions that they want their search results to match.
Faceted Result #2
Faceting allows the exact result that fits the user's criteria and goals to be surfaced
Applying filters makes unwieldy volumes of results manageable

Figure 5.1: An illustration of faceted search by Yonik Seeley. Source: [54].

5.1.2 Concept

Building upon the concept of faceted navigation, we designed our special form of visualization for our *MFA* such that we satisfy the objectives discussed before of an efficient, healthy exposure and smooth experience for human domain experts dealing with content of an unpleasant nature. We designed a simple and basic interface that presents the results of previous analysis and techniques to allow users to perform knowledge retrieval and extraction from these documents.

The introduced multifaceted interface has three facets (or in other words levels) such that the first facet, level 1, would display a list of documents sorted in ascending order based on the overall score, described in section 4.6, of each document alongside with some other features such as the topic title and username that we decided to include in this facet. The second facet, level 2, provides an extra in-depth and detailed features and attributes that are the result of all the techniques applied before on this document; the goal of this level is to provide enough summarization of the document content that would be sufficient to describe the document but without actually reading it yet or being exposed to its inflammatory content. The third and lowest facet, level 3, contains the actual raw and unprocessed content of the document. However, this facet would be accessed only if the human expert thought that based on the summarization in the previous level this document is worth reading it. Or it is important or critical and the summarization was not sufficient, so users got to read the actual content. Otherwise, the document's actual raw content is hidden.

Our idea and underlying motivation behind the interface design is to have content flow from abstract yet informative level — the overall score indicates to a certain degree the overall importance of each document — to the summarization of the document with all domain-related features to finally the actual raw detail and content. If you imagine an arrow moving along from each facet to another the arrow width would start at first with moderate width then it would increase to a broader one then it would increase extra more in the final level. The illustration below describes the idea.

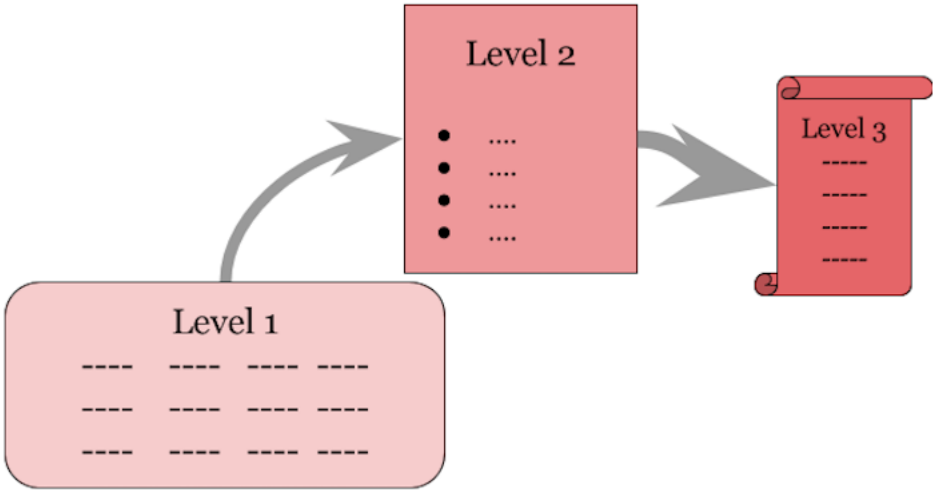
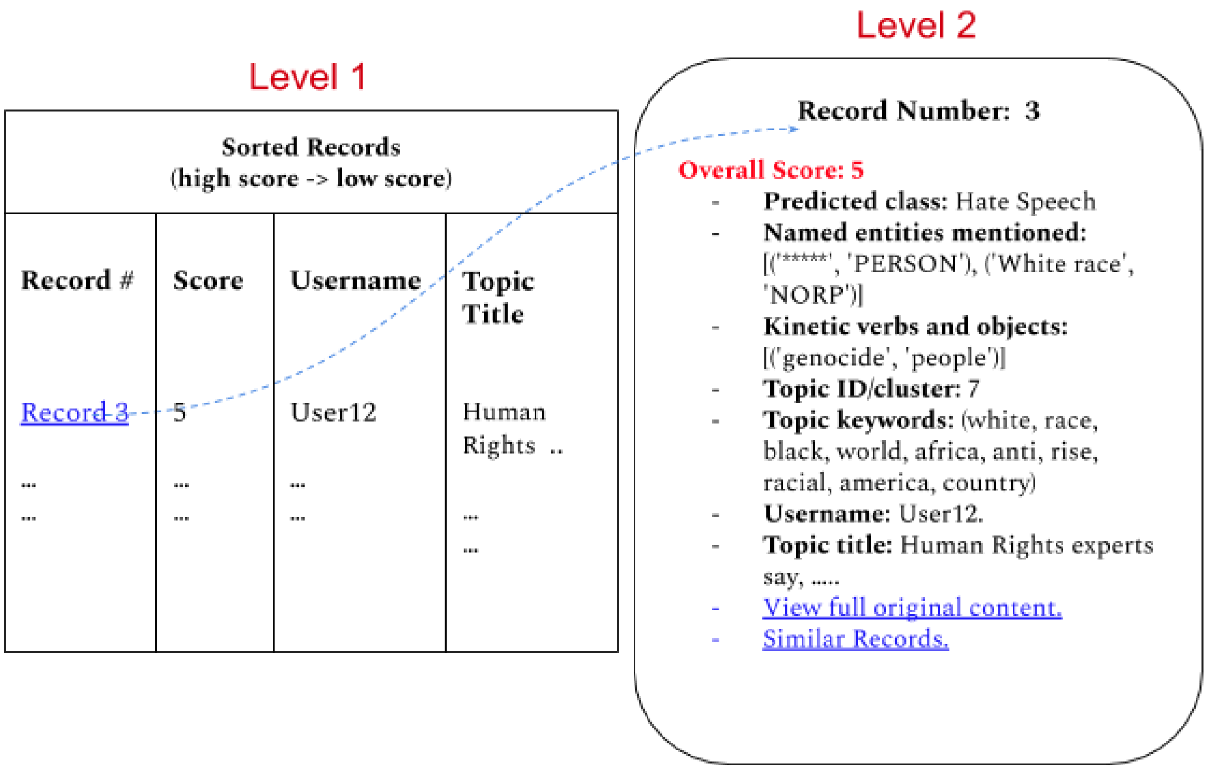


Figure 5.2: An illustration of level of detail idea in our *MFA* Approach.

Facets Levels Descriptions

- Document list (Level 1) is the first page of all the documents with document number, username, title sorted from highest to lowest score.
- Document summary (Level 2) is the page containing the score, predicted class, entities involved, kinetic actions found, possible targets.
- Document detail (Level 3) is the page containing the original, detailed, raw document, without any filters, blurred words or anything.

Facets Levels Illustrations



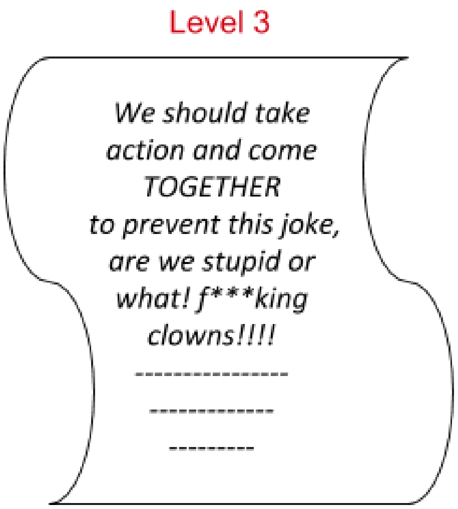
Sorted Records (high score -> low score)			
Record #	Score	Username	Topic Title
Record 3	5	User12	Human Rights ..
...	...	...	
...	...	...	...
			...

Record Number: 3

Overall Score: 5

- **Predicted class:** Hate Speech
- **Named entities mentioned:** [('*****', 'PERSON'), ('White race', 'NORP')]
- **Kinetic verbs and objects:** [('genocide', 'people')]
- **Topic ID/cluster:** 7
- **Topic keywords:** (white, race, black, world, africa, anti, rise, racial, america, country)
- **Username:** User12.
- **Topic title:** Human Rights experts say, ...
- View full original content.
- Similar Records.

Figure 5.3: Level 1 and Level 2 in our *MFA*.

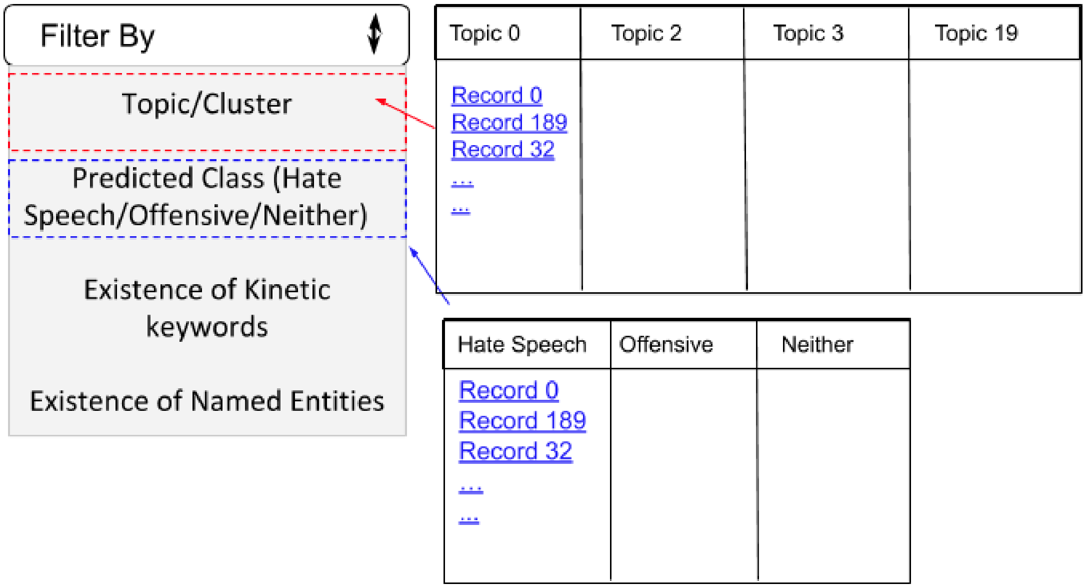


*We should take action and come TOGETHER to prevent this joke, are we stupid or what! f***king clowns!!!!*

Figure 5.4: Level 3 in our *MFA*.

Filtering and Similar Documents Options

Additionally, we have also provided two important features to help navigate and browse the documents. The first feature is the "Filter" feature, which can be based on criteria as the topic cluster, predicted class or kinetic keywords. The second one is the "Similar Records" feature that is based on measures of similarity between the currently viewed document and the other documents — we used topic modeling clusters and embeddings to calculate the similarity level. Below is an illustration of these two.

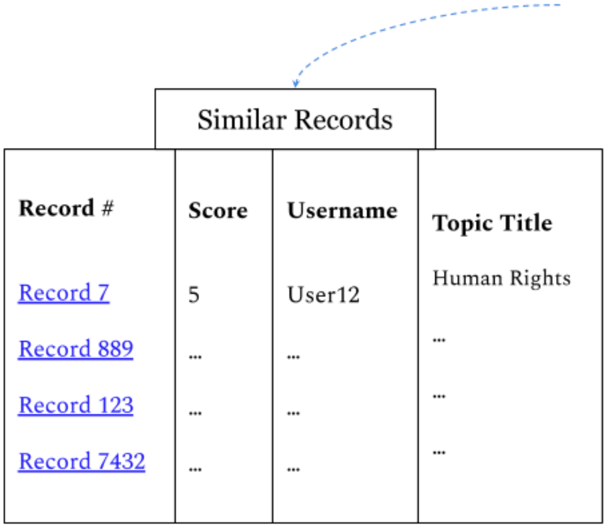


Filter By
Topic/Cluster
Predicted Class (Hate Speech/Offensive/Neither)
Existence of Kinetic keywords
Existence of Named Entities

Topic 0	Topic 2	Topic 3	Topic 19
Record 0 Record 189 Record 32			

Hate Speech	Offensive	Neither
Record 0 Record 189 Record 32		

Figure 5.5: Filtering Option in our *MFA*.



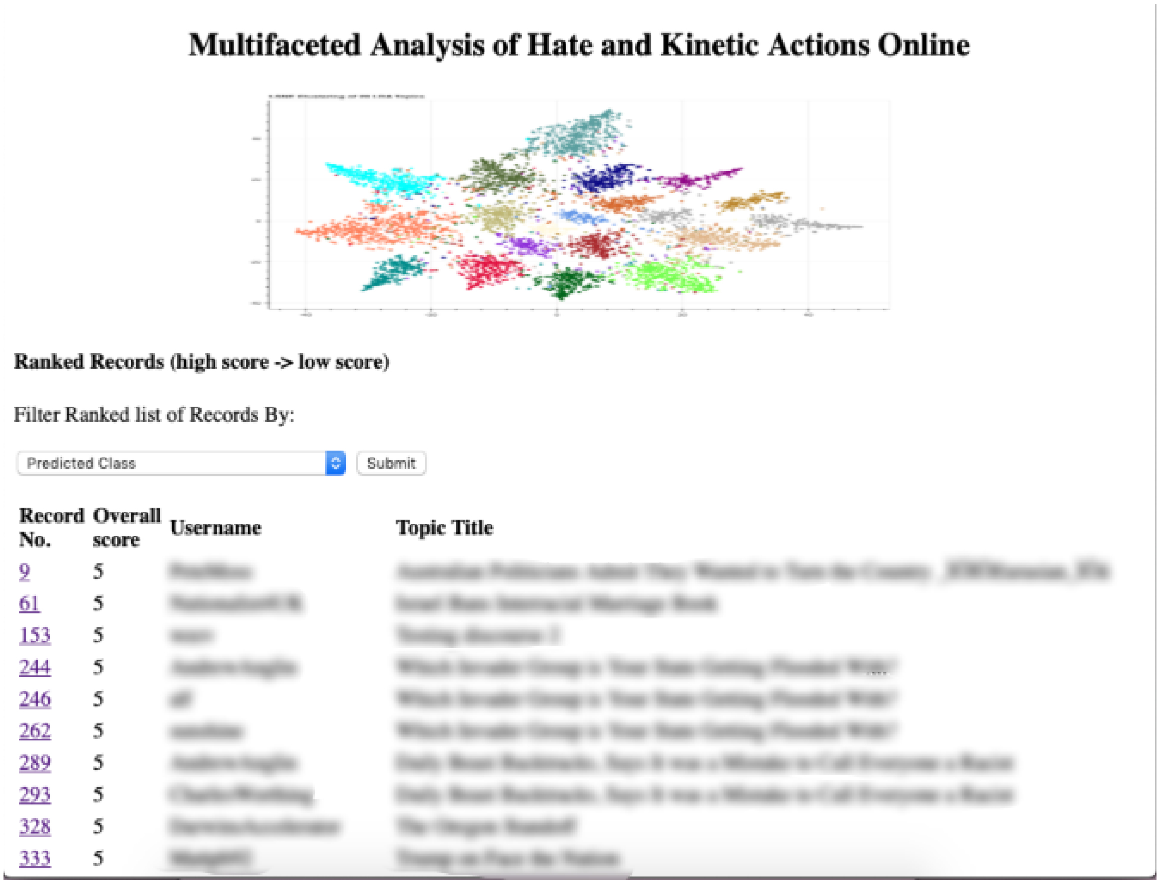
Record #	Score	Username	Topic Title
Record 7	5	User12	Human Rights
Record 889	...	...	...
Record 123	...	...	...
Record 7432	...	...	...

Figure 5.6: The similar documents option in our *MFA*.

5.2 Flask Web Application

To both illustrate and test the concept of our *MFA* approach, we have built a web application using FLASK, which is a web framework in Python [1]. We chose it because of its simplicity and easy to use functions as it does not require any third-party libraries. In this section, we will show some snippets of this web app to explain the ideas discussed before. We also used this web app to test and evaluate our approach which we will explain in detail in the next chapter. **Please note**, we have blurred some parts because of a data disclosure and confidentiality agreement so that we are able to provide an anonymous study without any identity information exposure.

In figure 5.7, you can see the first home page which is the first facet in our *MFA* or what we called, **Level 1**. The level displays the ranked documents along with each document score, username and topic title. Also, the filtering feature can be accessed in this level too by clicking on the drop-down menu provided.



Multifaceted Analysis of Hate and Kinetic Actions Online

Ranked Records (high score -> low score)

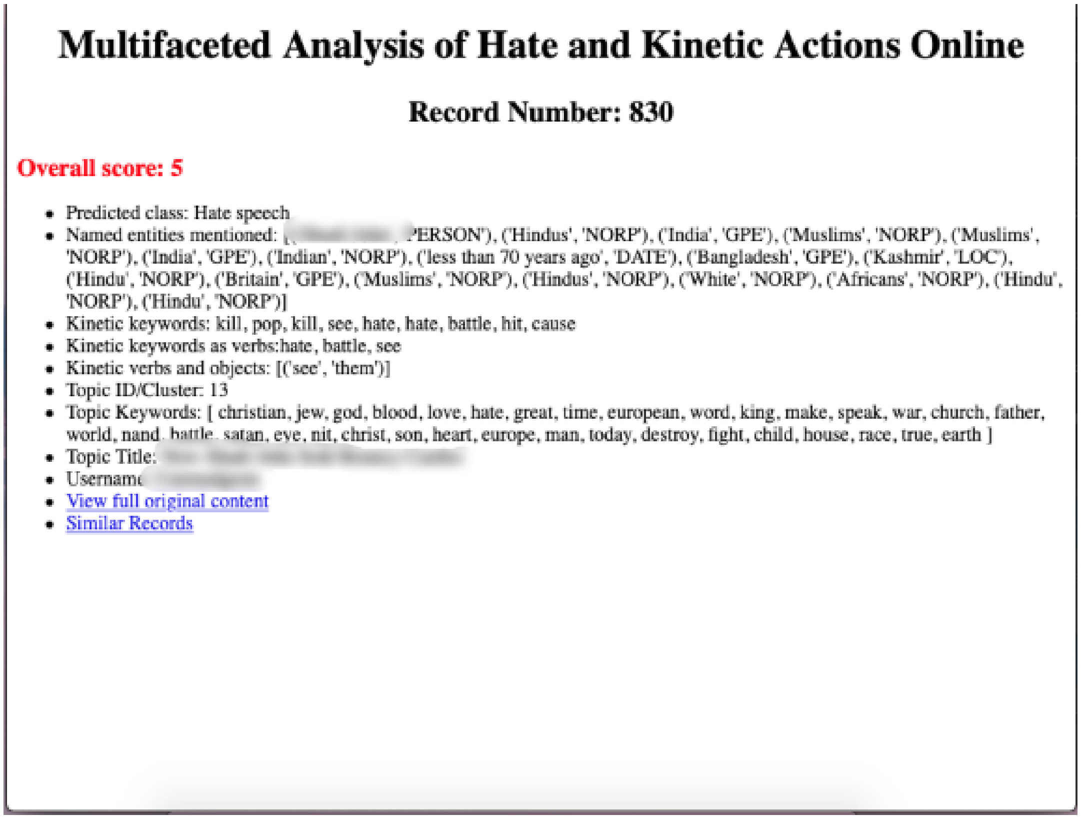
Filter Ranked list of Records By:

Record No.	Overall score	Username	Topic Title
9	5	[redacted]	[redacted]
61	5	[redacted]	[redacted]
153	5	[redacted]	[redacted]
244	5	[redacted]	[redacted]
246	5	[redacted]	[redacted]
262	5	[redacted]	[redacted]
289	5	[redacted]	[redacted]
293	5	[redacted]	[redacted]
328	5	[redacted]	[redacted]
333	5	[redacted]	[redacted]

Figure 5.7: Level 1 as in our *MFA*.

[1] `https://palletsprojects.com/p/flask/`, `http://flask.palletsprojects.com/en/1.1.x/`

If the domain expert user clicks on the record number in the first level, he/she can then have access to the second facet or **Level 2**. Figure 5.8 provides an example of the summarization in this level. The features include the overall score, predicted class, entities involved, kinetic keywords found, potential targets as in kinetic objects, topic cluster and topic's top keywords. In this facet, the domain expert user can move on to the third level by pressing on "View full original content" or check other documents which are similar to currently viewed one.



Multifaceted Analysis of Hate and Kinetic Actions Online

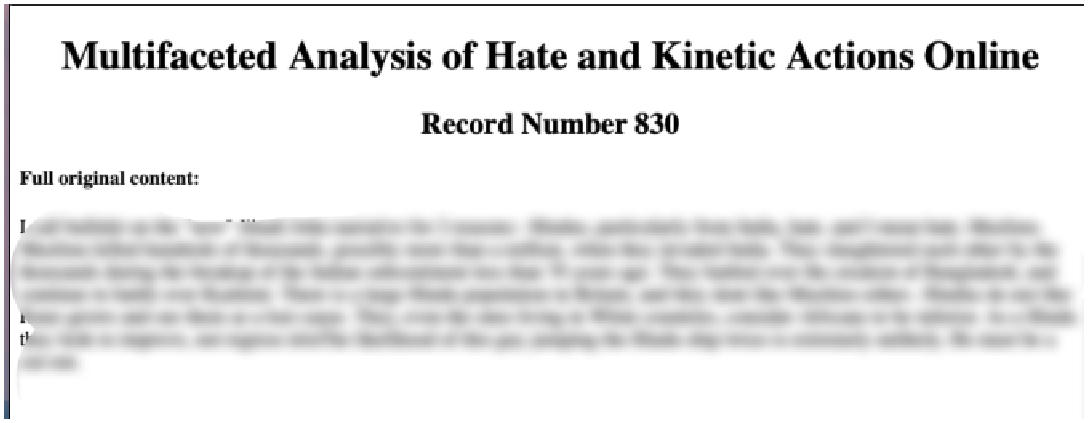
Record Number: 830

Overall score: 5

- Predicted class: Hate speech
- Named entities mentioned: [redacted] PERSON'), ('Hindus', 'NORP'), ('India', 'GPE'), ('Muslims', 'NORP'), ('Muslims', 'NORP'), ('India', 'GPE'), ('Indian', 'NORP'), ('less than 70 years ago', 'DATE'), ('Bangladesh', 'GPE'), ('Kashmir', 'LOC'), ('Hindu', 'NORP'), ('Britain', 'GPE'), ('Muslims', 'NORP'), ('Hindus', 'NORP'), ('White', 'NORP'), ('Africans', 'NORP'), ('Hindu', 'NORP'), ('Hindu', 'NORP')]
- Kinetic keywords: kill, pop, kill, see, hate, hate, battle, hit, cause
- Kinetic keywords as verbs:hate, battle, see
- Kinetic verbs and objects: [('see', 'them')]
- Topic ID/Cluster: 13
- Topic Keywords: [christian, jew, god, blood, love, hate, great, time, european, word, king, make, speak, war, church, father, world, nand, battle, satan, eye, nit, christ, son, heart, europe, man, today, destroy, fight, child, house, race, true, earth]
- Topic Title: [redacted]
- Usernam[redacted]
- View full original content
- Similar Records

Figure 5.8: Level 2 as in our *MFA*.

In the third facet, **Level 3**, the domain expert user can then have full access to the whole unprocessed, raw, unblurred and unstructured content. This content may include directed insults, swears, hate speech, threats, sarcasm, racial or religious or sexual language. Figure 5.9 provides an example of the level.



Multifaceted Analysis of Hate and Kinetic Actions Online

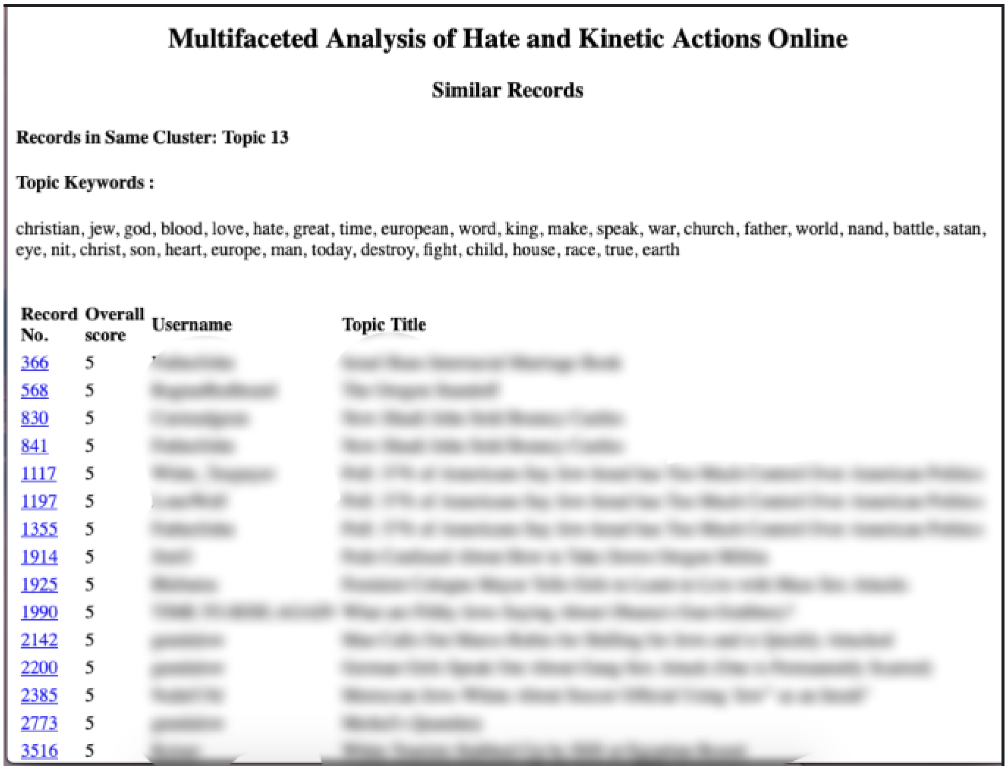
Record Number 830

Full original content:

[redacted]

Figure 5.9: Level 3 as in our *MFA*.

In figure 5.10, one can see the list of similar documents relative to the currently viewed document. The preview shows the same set of features displayed in level 1. These features can also be viewed in level 2. In figures 5.11 and 5.12, one can see how the filtering feature works. In this example, we chose the option to filter the documents by the predicted class. The remaining options are filtered by named entities existence, kinetic keywords or topic keywords.



Multifaceted Analysis of Hate and Kinetic Actions Online

Similar Records

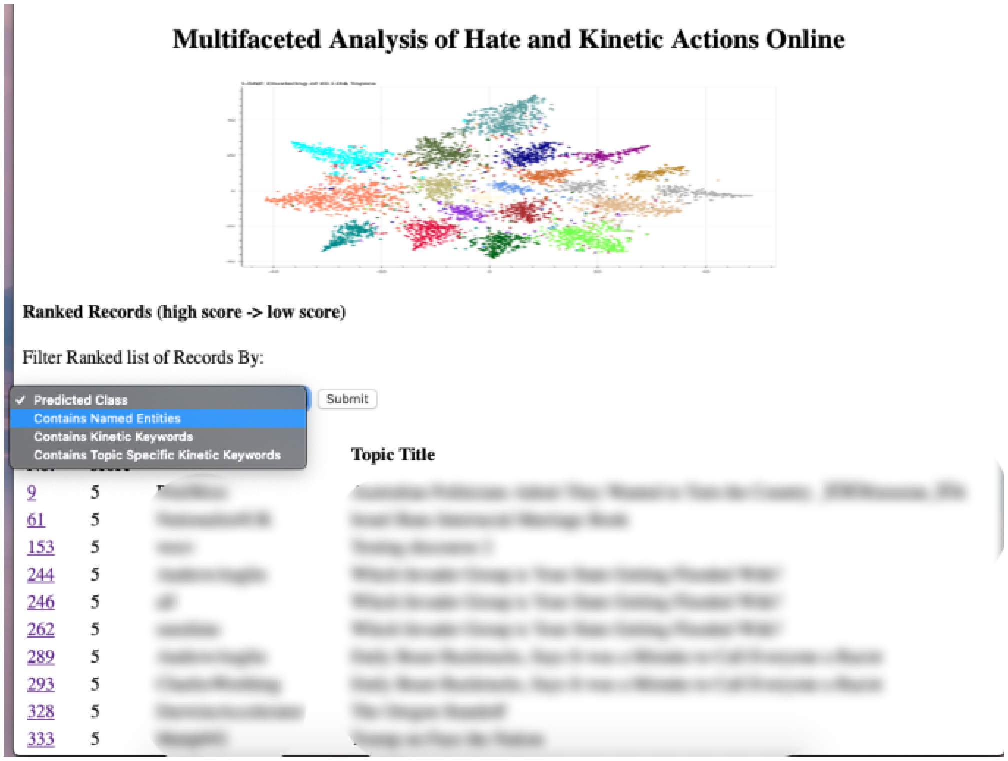
Records in Same Cluster: Topic 13

Topic Keywords :

christian, jew, god, blood, love, hate, great, time, european, word, king, make, speak, war, church, father, world, nand, battle, satan, eye, nit, christ, son, heart, europe, man, today, destroy, fight, child, house, race, true, earth

Record No.	Overall score	Username	Topic Title
366	5	[redacted]	[redacted]
568	5	[redacted]	[redacted]
830	5	[redacted]	[redacted]
841	5	[redacted]	[redacted]
1117	5	[redacted]	[redacted]
1197	5	[redacted]	[redacted]
1355	5	[redacted]	[redacted]
1914	5	[redacted]	[redacted]
1925	5	[redacted]	[redacted]
1990	5	[redacted]	[redacted]
2142	5	[redacted]	[redacted]
2200	5	[redacted]	[redacted]
2385	5	[redacted]	[redacted]
2773	5	[redacted]	[redacted]
3516	5	[redacted]	[redacted]

Figure 5.10: Similar Documents as in our *MFA*.



Multifaceted Analysis of Hate and Kinetic Actions Online

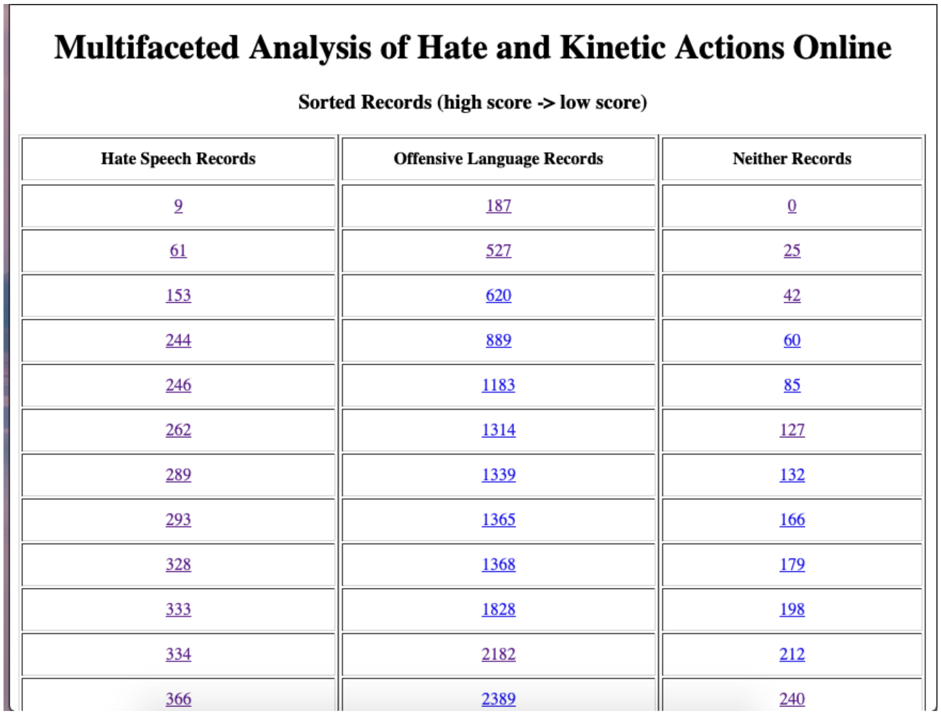
Ranked Records (high score -> low score)

Filter Ranked list of Records By:

✓ Predicted Class
Contains Named Entities
Contains Kinetic Keywords
Contains Topic Specific Kinetic Keywords
Submit

			Topic Title
9	5	[redacted]	[redacted]
61	5	[redacted]	[redacted]
153	5	[redacted]	[redacted]
244	5	[redacted]	[redacted]
246	5	[redacted]	[redacted]
262	5	[redacted]	[redacted]
289	5	[redacted]	[redacted]
293	5	[redacted]	[redacted]
328	5	[redacted]	[redacted]
333	5	[redacted]	[redacted]

Figure 5.11: Filtering Option as in our *MFA*.



Multifaceted Analysis of Hate and Kinetic Actions Online

Sorted Records (high score -> low score)

Hate Speech Records	Offensive Language Records	Neither Records
9	187	0
61	527	25
153	620	42
244	889	60
246	1183	85
262	1314	127
289	1339	132
293	1365	166
328	1368	179
333	1828	198
334	2182	212
366	2389	240

Figure 5.12: One of the filtering options as in our *MFA*.

Chapter 6

Analysis

Given the wide range of computational and visual techniques we have introduced into our system, it is important to establish some criteria for the evaluation of an analyst's workbench like the one we have provided. Traditionally, such systems are evaluated through user studies. But given the early stage of our work, which is still in the proof-of-concept stage, we wanted to introduce some guiding principle which can form the basis for future research, while at the same time give us guidance for continued development work.

6.1 Domain Expert Research Collaboration

To analyze our proof of concept approach, we held a research collaboration meeting with domain experts subjects and provided them with access to our system and the following background:

Overview

The system will be providing different ways of accessing documents. It will also be proving short summaries of documents so that analysts will not need to read all the documents. We are interested in whether by looking at a document summary, whether or not the analyst will be able to say, (based on this summary, I do not need to read the actual document).

Definition of MFA (Multifaceted Approach):

- Document list (Level 1) is the first page of all the documents with document number, username, title sorted from highest to lowest score.
- Document summary (Level 2) is the page containing the score, predicted class, entities involved, kinetic actions found, possible targets.
- Document detail (Level 3) is the page containing the original, detailed, raw document, without any filters, blurred words or anything.

For our meeting with the analyst, we propose the following steps:

1. Allow the analyst to scroll through the document list and randomly pick up to 20 documents to read from the documents assigned the highest possible score (cluster score=5).
2. The analyst will click on the document, and the system will provide the document summary.
3. After reading the summary, the analyst will decide whether to read the actual document. The full document is accessed by clicking on the link.
4. Repeat steps 1 through 3 above for the 20 lowest scoring documents (cluster score 0)

And asked them the following questions:

- By looking at the summary, how often is someone able to say (No I don't need to read the detailed document?)
- How much time he is spending on each level to gain the necessary knowledge?
- What value is each level giving to an expert user?

6.2 Discussion

We determined that future evaluation of a system based on these metrics could be:

- How relevant are the documents placed in highest or lowest scored cluster is to the overall severity of the document? In other words, does the approach do a good job by placing them at the top/bottom or not?
- Are the summary and knowledge provided at every level sufficient for the overall efficiency of the analyst's task?
- Does *MFA* make information query and retrieval easier when looking for a particular task?
- Overall, does the approach succeed in reducing the exposure to disturbing content while users are performing their daily work?

For the top-scored 20 random records, our performance analysis shows that the answer to the relevance question was positive, based on their experience in this field. The randomly picked documents deserve the overall score given by our approach and think they were relevant, in other words, it is good that they were placed at the top of the list in cluster for score 5. As for the 20 documents which were picked randomly from the cluster 0 (minimum score), these documents summary were not as insightful as the first group. This is due to the fact that most of the low scored documents did not contain much of a textual content. As they were mainly pictures, links or GIF [1] or even NAN values (empty content), which explains the low score assigned to them. However, low score does not necessarily mean that these were unimportant documents as users noticed sometimes the topic cluster, which some of these documents were in, was important or the username was someone known to them. But since this is a natural processing analysis study, the absence of textual content resulting in placing these records in the bottom was completely predictable and expected. In other words, they deserved the low score. One positive comment was that having most of the documents that are does not contain much of a text like empty or contain pictures placed in a separate cluster (in our case it is the score 0 cluster) served a lot in helping provide means for structuring the data and organizing it. Now, in a way they figured that images are at the bottom and are not mixed with the rich textual documents, thus, saving their time and searching efforts.

Let us move to the question of how sufficient the summary that was provided at each level. It was shown it was sufficient to decide whether or not to read the full content or to decide to move on to another document as users are usually looking for particular keywords, hints, usernames, entities and all of this was provided using the summary in a very clean and structured way. Now, since we were performing a demo on a particular task it was worth checking how would *MFA* do to achieve a successful and efficient information extraction — although that was not the main objective of the session — the answer was that it did indeed help and was useful when looking for a particular information. Finally, the main goal was to see if the *MFA* helped in redaction of unnecessary exposure to potentially harmful content and the answer was Yes indeed. It helped significantly decrease the amount of searching and digging into the unstructured data because, unlike, before users would have to read all of the documents to analyze it and then decide whether it is relevant, useful, important or not and about what/whom it is talking about. However, now users are having a processed, clean, structured data with an overall embedded score to indicate the importance, content richness or severity saving a lot of effort.

Additional evaluation of the system by the above mentioned domain experts is still in progress. While the textual data itself is confidential, we are obtaining metadata related to the textual data and more detailed information about how the experts process this data. The results from this additional evaluation will be available for disclosure in an upcoming publication.

To summarize, the multifaceted summarizing approach of having the content accessed on different levels and to have it described by multiple categories at the same time, was very efficient, accurate, helpful and reduced the time a particular task would normally require from two and a half days of going through the records to one or two hours with

[1] A lossless format for image files that supports both animated and static images.

the same anticipated outcome. So *MFA* helped with the problems of workload, data query, information extraction and the redaction of excessive harmful content exposure.

Chapter 7

Conclusion and Future Work

To conclude this thesis work, we have introduced a Multi-Faceted Approach ***MFA*** to help navigate, explore, describe and redact the exposure of documents of inflammatory nature. This approach is built using an application of various automatic techniques such as query expansion, LDA topic modeling, part of speech tagging, named entity recognition, hate speech detection model and information retrieval and visualization, and other human-engineered techniques such as seed list creation of domain-related keywords and the embedded overall score scheme design. The approach follows a pipeline that moves from each method to another while keeping track of all of the results. Such that by the end of this analysis, the unstructured data was converted into a structured and easy to interpret data.

We have provided a baseline and foundation that can be easily built upon and optimized. As future work, other facets of interest might be simply added to the pipeline to provide more features to be added to documents summarization. Also, the scoring scheme could be built on to include more attributes like for example records with a high number of replies, or to add more filters on existing attributes like records with violence-associated verbs linking to "name" person should be scoring as higher than name "country". Another possible work, is to expand this approach to include analysis of non-textual data as well like for example, include image processing to analyze the images in these documents and extract extra information from it to establish inflammatory content exposure redaction from these images content as well.

A final note on the usability of our approach. The domain experts —in particular CASIS-Vancouver— have validated the usability of this concept and approach on their daily informational retrieval and analyzing of the documents for their tasks. A demo task that we held during the research collaboration session was to look for any possible extremist recruiters targeting college students.

While we have been focused on toxic terminology, our approach could be applicable to other domains, perhaps even insurance and taxation documents which often contain sensitive information that might want to be withheld from certain viewers. These domains might benefit from the confidentiality and exposure control features that our visualization concept can provide.

Overall, further development is needed here because it is a study that revolves around users and their experiences dealing with sensitive content. Hence, as we witness nowadays the rise of organizations that provide approaches similar to our framework, like Hatebase[1]. It helps organizations and online communities detect, monitor and analyze hate speech [1], it also provides open data for research on hate speech that supports different languages and regions. For example, as a future work Hatebase's available research data can be used to help optimize our hate speech classification step or expand our own lexical resource. There is also another publicly available datasets like the one provided for a Kaggle competition by Jigsaw and Google to study the bias towards certain keywords that can occur when training a hate speech classifier as models can somehow associate certain keywords or entities with toxicity even if they occur in a toxic-free context [2]. The availability of such data resources is additional evidence of the importance of the problem and the need for structured and labeled data to support any future work.

[1] https://hatebase.org

[2] https://www.kaggle.com/c/jigsaw-unintended-bias-in-toxicity-classification/overview

Bibliography

[1] Hatebase org., accessed: 2019-11-23, https://hatebase.org/about.

[2] No hate speech youth campaign, https://www.coe.int/en/web/no-hate-campaign, 2013, accessed: 2019-09-27.

[3] Masayuki Asahara and Yuji Matsumoto. Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 8–15. Association for Computational Linguistics, 2003.

[4] Pinkesh Badjatiya, Shashank Gupta, Manish Gupta, and Vasudeva Varma. Deep learning for hate speech detection in tweets. In *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW '17 Companion, pages 759–760, Republic and Canton of Geneva, Switzerland, 2017. International World Wide Web Conferences Steering Committee.

[5] Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. Cloze-driven pretraining of self-attention networks. *arXiv preprint arXiv:1903.07785*, 2019.

[6] Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[7] Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel. An algorithm that learns what's in a name. *Machine learning*, 34(1-3):211–231, 1999.

[8] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[9] Andrew Borthwick, John Sterling, Eugene Agichtein, and Ralph Grishman. Exploiting diverse knowledge sources via maximum entropy in named entity recognition. In *Sixth Workshop on Very Large Corpora*, 1998.

[10] Eric Brill. A simple rule-based part of speech tagger. In *Proceedings of the Third Conference on Applied Natural Language Processing*, ANLC '92, pages 152–155, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[11] Pete Burnap and Matthew L Williams. Cyber hate speech on twitter: An application of machine classification and statistical modeling for policy and decision making. *Policy & Internet*, 7(2):223–242, 2015.

[12] Nan Cao, Jimeng Sun, Yu-Ru Lin, David Gotz, Shixia Liu, and Huamin Qu. Facetatlas: Multifaceted visualization for rich text corpora. *IEEE transactions on visualization and computer graphics*, 16(6):1172–1181, 2010.

[13] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *Acm Computing Surveys (CSUR)*, 44(1):1, 2012.

[14] Robert Dale, Hermann Moisl, and Harold Somers. *Handbook of Natural Language Processing*. Taylor & Francis, 2000.

[15] Aida Mostafazadeh Davani, Leigh Yeh, Mohammad Atari, Brendan Kennedy, Gwenyth Portillo-Wightman, Elaine Gonzalez, Natalie Delong, Rhea Bhatia, Arineh Mirinjian, Xiang Ren, and Morteza Dehghani. Reporting the unreported: Event extraction for analyzing the local representation of hate crimes, 2019.

[16] Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. Automated hate speech detection and the problem of offensive language. In *Eleventh international aaai conference on web and social media*, 2017.

[17] Nemanja Djuric, Jing Zhou, Robin Morris, Mihajlo Grbovic, Vladan Radosavljevic, and Narayan Bhamidipati. Hate speech detection with comment embeddings. In *Proceedings of the 24th international conference on world wide web*, pages 29–30. ACM, 2015.

[18] Maeve. Duggan. Online harassment, pew research center: Internet, science, accessed: 2019-09-27, https://www.pewinternet.org/2017/07/11/online-harassment-2017/, 2017.

[19] Robert Faris, Amar Ashar, Urs Gasser, and Daisy Joo. Understanding harmful speech online. *Berkman Klein Center Research Publication*, (2016-21), 2016.

[20] Christiane Fellbaum. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

[21] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics, 2005.

[22] Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *biometrics*, 21:768–769, 1965.

[23] Iginio Gagliardone, Danit Gal, Thiago Alves, and Gabriela Martinez. *Countering online hate speech*. Unesco Publishing, 2015.

[24] Thushan Ganegedara. Intuitive Guide to LDA Article, https://towardsdatascience.com/light-on-math-machine-learning-intuitive-guide-to-latent-dirichlet-allocation-437c81220158. accessed: 2019-10-15.

[25] Òscar Garibo i Orts. Multilingual detection of hate speech against immigrants and women in twitter at SemEval-2019 task 5: Frequency analysis interpolation for hate in speech detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 460–463, Minneapolis, Minnesota, USA, June 2019. Association for Computational Linguistics.

[26] Abbas Hanon AlAsadi, Hind Mohammed, Ebtesam Alshemmary, Moslem, and Muslim Khudhair. Hybrid k-means clustering for color image segmentation. 06 2015.

[27] Marti Hearst. *Search user interfaces.* Cambridge university press, 2009.

[28] Marti Hearst, Ame Elliott, Jennifer English, Rashmi Sinha, Kirsten Swearingen, and Ka-Ping Yee. Finding the flow in web site search. *Communications of the ACM*, 45(9):42–49, 2002.

[29] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pages 289–296, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc.

[30] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[31] James B Jacobs, Kimberly Potter, et al. *Hate crimes: Criminal law & identity politics.* Oxford University Press on Demand, 1998.

[32] Kyo Kageura. 1.4. 3 multifaceted/multidimensional concept systems. *Handbook of terminology management: Basic aspects of terminology management*, 1:119, 1997.

[33] Paige. Leskin. A new study found a link between the number of racist tweets and real-life hate crimes in 100 us cities., business insider, accessed: 2019-09-27, https://www.businessinsider.my/twitter-racism-hate-speech-linked-real-life-hate-crimes-study-2019-6/, 2019.

[34] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 28(2):129–137, 1982.

[35] Meili Lu, Xiaobing Sun, Shaowei Wang, David Lo, and Yucong Duan. Query expansion via wordnet for effective code search. In *2015 IEEE 22nd International Conference on Software Analysis, Evolution, and Reengineering (SANER)*, pages 545–549. IEEE, 2015.

[36] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[37] Melvin Earl Maron and John Larry Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the ACM (JACM)*, 7(3):216–244, 1960.

[38] Pushkar Mishra, Marco Del Tredici, Helen Yannakoudakis, and Ekaterina Shutova. Author profiling for hate speech detection, 2019.

[39] Pushkar Mishra, Helen Yannakoudakis, and Ekaterina Shutova. Tackling online abuse: A survey of automated abuse detection methods. *arXiv preprint arXiv:1908.06024*, 2019.

[40] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007.

[41] Liming Nie, He Jiang, Zhilei Ren, Zeyi Sun, and Xiaochen Li. Query expansion based on crowd knowledge for code search. *IEEE Transactions on Services Computing*, 9(5):771–783, 2016.

[42] Chikashi Nobata, Joel Tetreault, Achint Thomas, Yashar Mehdad, and Yi Chang. Abusive language detection in online user content. In *Proceedings of the 25th international conference on world wide web*, pages 145–153. International World Wide Web Conferences Steering Committee, 2016.

[43] Article on ListenData. Named entity recognition using python, accessed: 2019-10-17, https://www.listendata.com/2018/05/named-entity-recognition-using-python.html.

[44] Jessie Ooi, Xiuqin Ma, Hongwu Qin, and Siau Chuin Liew. A survey of query expansion, query suggestion and query refinement techniques. In *2015 4th International Conference on Software Engineering and Computer Systems (ICSECS)*, pages 112–117. IEEE, 2015.

[45] Nedjma Ousidhoum, Zizheng Lin, Hongming Zhang, Yangqiu Song, and Dit-Yan Yeung. Multilingual and multi-aspect hate speech analysis, 2019.

[46] Christos H. Papadimitriou, Hisao Tamaki, Prabhakar Raghavan, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '98, pages 159–168, New York, NY, USA, 1998. ACM.

[47] John Pavlopoulos, Prodromos Malakasiotis, and Ion Androutsopoulos. Deep learning for user comment moderation. *CoRR*, abs/1705.09993, 2017.

[48] Quentin Pleple. Topic coherence to evaluate topic models, accessed: 2019-10-13, http://qpleple.com/topic-coherence-to-evaluate-topic-models/.

[49] Lisa F. Rau. Extracting company names from text. *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, i:29–32, 1991.

[50] Michael Röder, Andreas Both, and Alexander Hinneburg. Exploring the space of topic coherence measures. In *Proceedings of the eighth ACM international conference on Web search and data mining*, pages 399–408. ACM, 2015.

[51] Sebastian Ruder. Summary of current state-of-art in the ner field, accessed: 2019-10-13, http://nlpprogress.com/english/named_entity_recognition.html.

[52] Joni Salminen, Hind Almerekhi, Milica Milenković, Soon-gyo Jung, Jisun An, Haewoon Kwak, and Bernard J Jansen. Anatomy of online hate: developing a taxonomy and machine learning models for identifying and classifying hate in online news media. In *Twelfth International AAAI Conference on Web and Social Media*, 2018.

[53] Niloofar Safi Samghabadi, Afsheen Hatami, Mahsa Shafaei, Sudipta Kar, and Thamar Solorio. Attending the emotions to detect online abusive language, 2019.

[54] Yonik Seeley. Faceted search with solr, accessed: 2019-10-13, https://lucidworks.com/post/faceted-search-with-solr/.

[55] Satoshi Sekine, Ralph Grishman, and Hiroyuki Shinnou. A decision tree method for finding and classifying names in japanese texts. In *Sixth Workshop on Very Large Corpora*, 1998.

[56] Sonit Singh. Natural language processing for information extraction. *arXiv preprint arXiv:1807.02383*, 2018.

[57] Tom De Smedt, Sylvia Jaki, Eduan KotzÃľ, LeÃŕla Saoud, Maja GwÃ§ÅždÅž, Guy De Pauw, and Walter Daelemans. Multilingual cross-domain perspectives on online hate speech, 2018.

[58] Sara Owsley Sood, Judd Antin, and Elizabeth Churchill. Using crowdsourcing to improve profanity detection. In *2012 AAAI Spring Symposium Series*, 2012.

[59] Ellen Spertus. Smokey: Automatic recognition of hostile messages. 1997.

[60] Stackabuse. Python for nlp part of speech and named entity recognition, accessed: 2019-10-17, https://stackabuse.com/python-for-nlp-parts-of-speech-tagging-and-named-entity-recognition/.

[61] Betty van Aken, Julian Risch, Ralf Krestel, and Alexander Löser. Challenges for toxic comment classification: An in-depth error analysis. *arXiv preprint arXiv:1809.07572*, 2018.

[62] Samuel Walker. *Hate speech: The history of an American controversy.* U of Nebraska Press, 1994.

[63] William Warner and Julia Hirschberg. Detecting hate speech on the world wide web. In *Proceedings of the Second Workshop on Language in Social Media*, LSM '12, pages 19–26, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[64] Matthew L Williams, Peter Burnap, Han Liu, Amir Javed, and Abdullah Ozalp. Hate in the machine: Anti-black and anti-muslim social media posts as predictors of offline racially and religiously aggravated crime. *British Journal of Criminology*, 2019.

[65] Jayani Withanawasam. *Apache Mahout Essentials Book. Clustering Chapter. Hard and Soft Clustering.* 2015.

[66] Jiewen Wu, Ihab Ilyas, and Grant Weddell. A study of ontology-based query expansion.