PRIVACY PRESERVING DATA PUBLISHING FRAMEWORK FOR UNSTRUCTURED TEXTUAL SOCIAL MEDIA DATA

Peruma Baduge Prasadi Apsara Abeywardana

(189302A)

Dissertation submitted in partial fulfillment of the requirements for the degree Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

July 2020

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works.

---------- ----------

P.B.P.A. Abeywardana Date

I certify that the declaration above by the candidate is true to the best of my knowledge. The above candidate has carried out research for the Masters dissertation under my supervision.

---------- ----------

T. Uthayasanker (PhD) Date

Abstract

Privacy has become an essential part of data science and analytics due to the potential of personal data misuse. As a result of privacy breaches reported in various analytical studies privacy preservation has become a legal responsibility rather than a simple social responsibility. Preserving privacy of unstructured data is more challenging compared to structured data. Social media has become largely popular over the past couple of decades and they are pumping a huge amount of data at a high velocity into analytical systems. Social media profiles contain a wealth of personal and sensitive information, creating enormous opportunities for third parties to analyze them with different algorithms, draw conclusions and use in disinformation campaigns and micro targeting based dark advertising. The primary goal of this study is to provide a mitigation mechanism for privacy breaches happening via disinformation campaigns that are done based on the insights extracted from personal/sensitive data analysis. Specifically, this research is aimed at building a privacy preserving data publishing framework for unstructured and textual social media data without compromising the true analytical value of those data. A novel way is proposed to apply traditional structured privacy preserving techniques on unstructured data. Creating a comprehensive twitter corpus annotated with privacy attributes is another objective of this research, especially because the research community is lacking one.

An easily extensible framework that can be adopted by many domains is implemented here, integrating different concepts from the literature. A comprehensive set of experiments are also performed in order to assess the capabilities of the machine learning models, algorithms as well as to simulate some real-world privacy preserving data publishing use cases.

ACKNOWLEDGMENTS

I would like to express my gratitude to my advisor, Dr. T. Uthayasanker, for his invaluable support, inspiration, supervision, and useful suggestions throughout this research work. He was never reluctant to guide me through composing this report in a successful manner.

Also, I would like the thank my family who was very supportive throughout the process, specially my parents and my husband. And also, I am grateful to my friends and colleagues with whom I discussed the concepts and who gave me back valuable inputs and feedback.

I must thank the independent data annotators Sriyoukan Sriranjan and Lavanaraj Sivarasa, who helped me annotating the data corpus.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

API	Application Programming Interface
EU	European Union
EEA	European Economic Area
BDSG	Bundesdatenschutzgesetz
GDPR	General Data Protection Regulation
PPDP	Privacy Preserving Data Publishing
GSR	Global Science Research
PPDM	Privacy Preserving Data Mining
MinGen	Minimum Generalization Algorithm
PPDC	Privacy Preserving Data Collection
GPS	Global Positioning System
LM	Loss Metric
DM	Discernibility Metrics
IoT	Internet of Things
UK DA	United Kingdom Data Archive
KL Distance	Kullback-Leibler Distance

CHAPTER 1: INTRODUCTION

In an era, where a huge volume of data is generated each second from various sources and pumped into analytical systems with a high velocity, personal information revealing an individual's privacy can too be exposed to data science platforms.

1.1 Personal data

Mobile data, health care data, social media data and web usage data are a few domains which can pump a huge amount of personal data into analytical systems without the knowledge or the consent of individuals. Mobile devices being a day to day necessity of every individual, there is a high possibility in gathering data through mobile phones. Recognizing an individual using his mobile data has become relatively easy. Health care systems being digitized has given many benefits to the patients, but at the same time they can lead to severe breaches of privacy too. Then again web and the Internet has exploited a vast amount of data about individuals which can be used to personally identify someone based on their web usage patterns. There is one last area, which has reformed the sharing of personal information, that is none other than social media. People choose to share many information about themselves as well as their close ones, compromising the privacy of both parties [4].

Availability of this kind of data has created a huge pool of opportunities for data scientists, but at the same time they have created some restrictions too. So, the expectation from data scientists is to extract the real value from these data, without breaching the privacy, which is a tough task as it sounds. Any model which runs on top of these data should adhere to privacy regulations, at the same time it should not deviate from the intended analytical purpose of the model.

Attributes related to personal data can be classified as follows based on how they can identify an individual. These attributes are extracted and used in Privacy Preserving Data Publishing (PPDP) techniques [4].

1.1.1 Personal information identifiers

These are the attributes such as ID, name or email address that can be directly used to identify an individual. These attributes uniquely recognize individuals from others.

1.1.2 Quasi identifiers

These are the attributes that can be linked with other outside data and used to distinguish an individual. For instance, age, gender, profession, race, religion can be considered as quasi identifiers. These are not unique identifiers by themselves but can be combined with another set of quasi identifiers to uniquely recognize a person.

1.1.3 Sensitive attributes

These are the attributes that individuals do not want to reveal about themselves. Examples can be salary, relationship statuses and diseases.

1.1.4 Non- sensitive attributes

These are the attributes other than the above mentioned three types. They may not have a direct or indirect relationship to identify individuals.

1.2 Personal data in social media

Social media plays a vital role in today's data science and analytics. They create a huge lake of data for analysts with very high analytical value. At the same time, social media data contains a wealth of personal information too. Social media data can be of a very high variety including formats such as structured, unstructured and graph data. All of these different formats can contain previously mentioned categories of personal information. For example, consider a tweet which belongs to the category of unstructured data as follows.

"My **teacher** ***Mary***, who lived in ***Corktown*** died of ***cancer*** yesterday at age ***65***"

If we look at the above tweet, it reveals some sensitive information about a person named Mary by disclosing she had cancer. And if we look carefully and compare

different words in this tweet with our previous definitions of privacy attributes, we can see there are different quasi identifiers like job, region and age within this tweet too. So, preserving privacy of unstructured social media data involves identifying and anonymizing all these pieces of information.

Social media data is highly used for very important predictive analysis tasks which can derive many useful insights. But at the same time, they can be used for disinformation campaigns too using personal information and targeting individuals without their consent.

1.2.1 Social threats of personal data analysis

Social platforms offer their data to third parties and advertisers to use in their analysis and campaigns. But sometimes these data are used in micro targeted disinformation campaigns to share dark ads. These highly personalized adverts are heavily used in political and other contexts to influence individuals by sharing misinformation. To host micro targeted ad campaigns, a lot of information related to individuals, their preferences and personality are required, and social media undoubtedly contain a fortune of such data. In the recent incident that involved Facebook, Cambridge Analytica and Global Science Research (GSR), millions of US Facebook users' data were analyzed without their consent and used in voter targeting, which is unethical as it sounds [1]. A solution to these concerns might be a law enforced privacy preserving middleware that has to be adopted by any social media platform, before publishing their data to a third party.

As a result of this kind of incidents privacy preserving data publishing has become a legal responsibility than a mere social responsibility. Many legalities are formed around personal data privacy because of that.

1.3 Data protection regulations

Following are some of such novel legal requirements which arouse recently.

1.3.1 General Data Protection Regulation (GDPR)

This is a regulation imposed by European Union (EU) law on data protection and privacy for all individuals within the EU and the European Economic Area (EEA) [2]. This is applicable to exporting and processing personal data in a region outside EU as well. The intention of this regulation is to make it easy for non-European companies to work with European bodies without any data breaches.

1.3.2 Russian Federal Law on Personal Data

This is a regulation which emphasizes on systemizing the data processing of individuals in Russia. This emphasizes on localizing personal data of Russian citizens to Russia [3].

1.3.3 German Bundesdatenschutzgesetz (BDSG)

This governs the exposure of personal data, which are manually processed or stored in IT systems. This was being modified with certain amendments for a long period of time has become stricter in the recent past.

By considering the above facts, it is quite evident that privacy preservation is becoming a mandatory prerequisite in the data science lifecycle.

1.4 Motivation

Publishing sensitive data related to individuals in a way that protects their privacy was a topic of interest for some time and many techniques are implemented with the contribution from various disciplines such as social science, computer science, and statistics. Most of the research works are carried out related to structured data [6][7][8][9][10][11][12][13][14][15]. Majority of the research carried out in unstructured data are done in the medical domain [30][43][44]. And there a few other research works done in the web log anonymization domain related to unstructured data [45][46]. When it comes to social media, the main focus has been anonymizing the graph-based relationships [31], but not anonymizing the actual content in social media posts. There was no publicly available research that was focused to preserve the

privacy of unstructured/textual social media content preserving the usefulness of data at the same time. Therefore, combining PPDP concepts to come up with a stable framework which can sanitize data before the analytical model is applied, can be of utmost importance. Making such a framework easy to extend can improve the adaptability by different parties. The sanitization process becomes difficult if the data are unstructured. Separating or de-identification of sensitive data is also a hard task as it is subjective and indirect. This research aims at coming up with a privacy preserving middleware for unstructured data which can sanitize data before applying the analytical model in a way the real value of the analysis is not compromised.

1.5 Problem statement

Based on the above motivational factors the problem statement can be mentioned as follows.

“How to preserve the privacy of unstructured/textual social media data, before publishing to a third party for analysis, maintaining the utility of data too?”

1.6 Research objectives

This research was scoped to achieve the following objectives which are centered around the previously mentioned problem statement.

- To implement a framework that can be utilized to build PPDP (Privacy Preserving Data Publishing) pipelines for unstructured data with Twitter as a use case
    - Privacy Data Extraction
    - Privacy Data Anonymization
    - Anonymization Evaluation
    - Utility Evaluation
- To produce a twitter corpus tagged with privacy attributes that can be used by the research community
- To achieve more than 60% accuracy in privacy attribute extraction
- To keep the usability reduction less than 5% after anonymization

- To provide a way to integrate different steps into PPDP pipelines in a flexible way (Plug and Play Mechanism)
- To make the implementation easy to extend in the future
- Provide a set of benchmark measures on the system to be used by anyone willing to adapt a similar approach

1.7 Outline

In this introductory chapter a foundation was laid for the rest of the discussion, highlighting the importance of privacy preservation and some key terminology associated with the subject. The rest of the report will be organized as follows. Chapter 2 will discuss some literature in the area and Chapter 3 will talk about the approach followed in building the proposed solution. Chapter 4 will initiate a discussion on the experimental design and experimental results whereas Chapter 5 will conclude the dissertation highlighting research outcomes, limitations, and future work.

CHAPTER 2: LITERATURE REVIEW

A comprehensive literature survey was performed in order to get an adequate understanding of the work already carried out on this area. This section is organized into a set of subtopics covering different aspects of privacy preserving data mining and quoted with relevant examples from the literature.

As Mendes et al. discuss, privacy preservation can be done at different phases of the data mining process [5]. With the assumption that the entity gathering data is not trustworthy, privacy should be trusted at ***data collection time***. Data collected for a separate intention can be repurposed to achieve different analytical objectives. Publishing data to be used publicly or by third parties can be a requirement in such scenarios and the privacy must be preserved and sensitive data should not be compromised in this process. So, privacy preservation at ***data publishing time*** is another important aspect and that will be the primary focus of this research. Techniques applied at this phase is known as Privacy Preserving Data Publishing (PPDP) techniques. Finally, ***data mining output*** should also be ensured with privacy restrictions as that too can be very revealing.

2.1 Existing PPDP techniques

Many research works have been carried out to come up with various sanitization techniques to protect personal data at the data publishing phase. All these techniques and algorithms introduce a tradeoff between privacy and utility. To suit a specific scenario, finding the right sanitization technique which satisfies the privacy criterion and obtains the maximum possible utility can be very challenging. A couple of deterministic sanitization techniques are discussed below, quoting examples from the literature.

2.1.1 Suppression

This mechanism replaces some attribute values by a symbol like '*' to indicate those attributes are repressed. For instance, a credit card number can be suppressed as 34** **** ****. This is one of the simplest forms of sanitization. Suppression has been

used as a building block of many methodologies proposed in the privacy preserving data mining research area. Usha et al. [6] use suppression to obfuscate quasi identifiers in the clustering based non-homogeneous anonymization system they are proposing. They claim that their system ensures high utility, as they cluster the data based on sensitivity of attributes and quasi identifiers in each cluster is anonymized separately using suppression. And this suppression-based anonymization is varied based on the sensitivity level of each cluster. Kaur [7] proposes a hybrid approach for privacy preserving data mining using suppression and perturbation, which is intended to protect customer data of an online shopping business, maintaining the utility at the same time. Kumari et al. [8] uses suppression as one of the mechanisms to obfuscate data in the medical databases in their study on analyzing and performing PPDM on medical databases. Sweeny [9] proposes a novel algorithm called MinGen (The Minimal Generalization Alogrithm) which combines generalization and suppression to achieve k-anonymity with minimal distortion.

2.1.2 Generalization

This implies replacing an attribute with a generalized value of its class. Intervals can be used to represent numeric values whereas hierarchy definitions are needed for categorical values. For instance, male and female values of the gender attribute or a nationality attribute can be replaced with ‘Any’ which is a more general value. Generalizing the values ‘engineer’ and ‘artist’ in the ‘occupation’ column to the value ‘professional’ is another example. And the age vale 52 can be generalized to a range of [50,60]. Generalization makes sure that a combined set of quasi identifiers cannot be used to uniquely identify a person after generalizing. Generalizing would make some attributes/records identical and difficult to distinguish when projected on top of quasi identifiers. Zhang et al. [10] utilize generalization in a novel Privacy Preserving Data Collection (PPDC) algorithm, highlighting the fact that even though generalization is highly utilized in PPDP, it is not widely used in PPDC. Hajian et al. [11] come up with another algorithm which enables privacy preservation and discrimination prevention in data mining, using generalization as the main sanitization tactic. Yu et al. [12] suggest a multi attribute generalization strategy stepping forward

from the typical single attribute generalization methodology. Wong et al. [13] propose that following a non-homogeneous generalization approach giving different generalized values to records within a partition can reduce the anonymization error.

2.1.3 Swapping

As the name implies this includes swapping some attribute values. For example, swapping the gender values of two records. Hasan et al. [14] come up with a value swapping algorithm which they claim to preserve the privacy as well as the utility on top of a sliced data table. US Census Bureau uses data swapping as one of the major disclosure limitation or masking technique apart from other masking techniques such as release of data for only a sample of the population, limitation of detail and top bottom coding [15]. Feinberg et al. [16] modify the originally proposed data swapping mechanism in their study to enhance the privacy as well as the utility.

2.1.4 Anatomization

This involves separating quasi identifiers and sensitive attributes into different tables so that the relationship among them will be broken. Values will not be changed in anatomization. Susan et al. [17] bring up a privacy preservation approach for multiple sensitive attributes by combining anatomization with slicing. Marathe et al. [18] employ anatomization with enhanced slicing to protect privacy of health care data. They claim this approach can be applied to any number of sensitive attributes and can ensure the utility at the same time. Oksvort [19] reasons out in his survey about privacy preservation techniques that anatomization is better at preserving the utility of data as this mechanism does not change the true values of data.

2.1.5 Permutation

This is about creating groups or buckets based on quasi identifiers and then shuffling the values of their respective sensitive attributes in each group to break the relationship between quasi identifier and the sensitive attributes.

2.1.6 Perturbation

This is about replacing the original values of some sensitive attributes using some fake values. Adding random noise from a known distribution to the original data can also be considered as a perturbation technique [5]. Thuraisigham et al. [20] present a novel approach based on perturbation where the users can choose the privacy level they desire. Patel et al. [21] summarize a couple of data perturbation techniques such as noise additive perturbation, condensation-based perturbation, random projection perturbation and geometric data perturbation. While noise additive perturbation being the most commonly used perturbation technique, condensation-based perturbation aims at preserving the covariance matrix for multiple columns. Focus of random projection perturbation is to project a set of data points from one multidimensional space to another space whereas geometric data perturbation consists of a sequence of random geometric transformations [21].

The following tables show how a table of health records look, before and after applying a few privacy preservation techniques. Techniques like generalization and suppression is applied on these data.

Name	Age	Gender	Zip Code	Nationality	Disease
John	28	M	13053	Russian	Heart Disease
Jack	29	M	13055	Chinese	Heart Disease
Bruce	22	M	13061	Japanese	Heart Disease
Ann	`24	F	14332	Russian	Heart Disease
Lewis	41	M	14556	American	Cancer
Richard	45	M	13227	American	Cancer

Anders	50	M	13226	American	Cancer
Paul	37	M	13221	American	Flu
Janet	34	F	13229	American	Flu
Cary	56	M	13225	American	Flu

Table 2.1: Health Records

Name	Age	Gender	Zip Code	Nationality	Disease
John	20-29	Any	130**	Any	Heart Disease
Jack	20-29	Any	130**	Any	Heart Disease
Bruce	20-29	Any	130**	Any	Heart Disease
Ann	20-29	Any	14***	Any	Heart Disease
Lewis	40-59	Any	14***	Any	Cancer
Richard	40-59	Any	1322*	Any	Cancer
Anders	40-59	Any	1322*	Any	Cancer
Paul	30-39	Any	1322*	Any	Flu
Janet	30-39	Any	1322*	Any	Flu
Cary	40-59	Any	1322*	Any	Flu

Table 2.2: Anonymized Health Records

2.2 Existing privacy models

All these PPDP techniques are mainly built on top of a couple of privacy models. These models assume certain aspects/properties of the data being processed and apply

suitable techniques depending on those properties. These privacy models act as definitions which formalize the expectations of the term ‘privacy’, otherwise it can be very subjective.

The term ‘***disclosure risk***’ is very important in any discussion built around privacy definitions [22]. It refers to the measurable estimate of a likelihood towards a privacy crack. So, in order to say that the privacy is preserved by applying above mentioned techniques, disclosure risk should be measured and quantified. There is a huge pool of research work carried out on measuring the disclosure risk and that will be touched towards the latter part of this literature review.

Following are three popular privacy models or definitions on top of which previously mentioned techniques are built.

2.2.1 k-anonymity

Proposed by Samarati and Sweeny [23], k-anonymity is the most extensively adapted privacy definition. A data set has k-anonymity property if the data for each individual cannot be eminently differentiated from at least k - 1 other persons who reside in the same dataset. In the original paper, they mention about a type of attack called linking attack, which can link information from more than one source to personally identify an individual. Therefore, even if we remove uniquely identifiable information from one source, they can be joined with another source to identify the person based on quasi identifiers. This kind of attacks need an extensive privacy model and techniques rather than mere anonymization. To protect information from this kind of linking attacks Samarati and Sweeny present k-anonymity.

For example, Table 2.3 is 4-anonymous. We cannot separate Ann’s record from 3 other records.

Name	Age	Gender	Zip Code	Nationality	Disease
Ann	20-29	Any	130**	Any	Heart Disease

Bruce	20-29	Any	130**	Any	Heart Disease
James	20-29	Any	130**	Any	Viral Infection
Janet	20-29	Any	130**	Any	Viral Infection
Fox	40-59	Any	14***	Asian	Cancer
Richard	40-59	Any	14***	Asian	Flu
Anders	40-59	Any	14***	Asian	Cancer
Paul	40-59	Any	14***	Asian	Flu
Helen	30-39	Any	1322*	American	Cancer
Cary	30-39	Any	1322*	American	Cancer
John	30-39	Any	1322*	American	Cancer
Jack	30-39	Any	1322*	American	Cancer

Table 2.3: 4-Anonymized Health Records

With k-anonymity in place, whatever the public database the adversary has access to, he cannot join and distinguish Ann, from the other 3 similar records. In the original paper, generalization and suppression are used as the main techniques to achieve k-anonymity. The anonymity requirement is expressed by identifying quasi identifiers and specifying the value of k (minimum duplicate records). Specifying the correct set of quasi identifiers and the k value can be very challenging.

Chen et al. [22] points out that there are some issues with k-anonymity that should be addressed by a more enhanced model. For instance, in table 3, last four individuals have cancer. So, if an adversary knows the age, gender, zip code and nationality of Jack from another public data source, he can easily tell Jack has cancer.

Despite of its limitations, many research works has been carried out making k-anonymity model as the base privacy definition. Arribas et al. [24] use k-anonymity for privacy preserving data mining of query logs. Their proposed method ensures the k-anonymity of query logs using micro-aggregation. Ni et al. [25] highlight that typical k-anonymity with local generalization is computationally expensive and they suggest a clustering-based k-anonymity approach to overcome these challenges. In order to support large data sets, they introduce parallelization to their novel k-anonymity algorithm.

2.2.2 l-diversity

This is an extension to the k-anonymity model, which diminishes the granularity of data using mechanisms including generalization and suppression. This tries to overcome a couple of weak points of the k-anonymity model. K-anonymity mostly provides the protection against linking attacks. But Machanavajjhala et al. [26] prove that there are two types of attacks that can break k-anonymity definition. If the sensitive attributes have a little diversity, an attacker can easily determine the values. Secondly, they argue that, if the attacker has background knowledge about the data, then again, they can break the k-anonymity privacy definition. As a solution to these two types of breaches, they are proposing the variation 'l-diversity'.

The first type of attack is mentioned while discussing the k-anonymity problem, where we can identify some attributes of a person, without uniquely identifying him. For example, according to Table 2.3, if someone knows John's age, gender, and zip code, he will probably know that John has cancer. This type of attacks or breaches are called homogeneity attacks [22]. And also, if someone has background knowledge about the data, they can infer the correct result even the records show some sort of variability. This is called background knowledge attacks.

To support the above two types of attacks Machanavajjhala et al. [26] come with the l-diversity principle. A table is said to satisfy the l-diversity principle if every group of tuples that share the same quasi identifier values in the table have at least l well-represented sensitive values; i.e., there are at least l-distinct sensitive values that are of roughly equal proportion [26]. An equivalence class is known to be a class which

contains similar records after anonymizing their quasi identifiers. L-diversity is considered in the context of an equivalence class.

Name	Age	Gender	Zip Code	Nationality	Disease (Sensitive Attribute)
Ann	20-29	Any	130**	Any	Heart Disease
Bruce	20-29	Any	130**	Any	Heart Disease
James	20-29	Any	130**	Any	Viral Infection
Janet	20-29	Any	130**	Any	Cancer
Fox	40-59	Any	14***	Asian	Cancer
Richard	40-59	Any	14***	Asian	Flu
Anders	40-59	Any	14***	Asian	Cancer
Paul	40-59	Any	14***	Asian	Heart Disease
Helen	30-39	Any	1322*	American	Cancer
Cary	30-39	Any	1322*	American	Flu
John	30-39	Any	1322*	American	Viral Infection
Jack	30-39	Any	1322*	American	Viral Infection

Table 2.4: 3-Diverse Health Records

According the above table, we can see that within each anonymous group, there are at least 3 different values for the sensitive attribute.

Authors of the original l-diversity model claim that l-diversity is more practical and addresses most of the inadequacies of k-anonymity. And it solves both homogeneous attacks and background knowledge attacks that can be caused by adversaries.

2.2.3 t-closeness

This is an enhancement to l-diversity model to overcome its flows. Even though l-diversity mitigates most of the issues prevailing in k-anonymity, it has its own weaknesses too. For instance, similarity attacks are a common type of attacks that adversaries can take advantage over l-diversity.

For instance, consider the following anonymized table which contains the salary and diseases of a set of individuals as sensitive attributes.

Zip Code	**Age**	**Salary**	**Disease**
476**	2*	3K	Gastric ulcer
476**	2*	4K	Gastritis
476**	2*	5K	Stomach cancer
4790*	>40	6K	Gastritis
4790*	>40	11K	Flu
4790*	>40	8K	Bronchitis
476**	3*	7K	Bronchitis
476**	3*	9K	Pneumonia
476**	3*	10K	Stomach cancer

Table 2.5: 3-Diverse Anonymized Health/Salary Records

If an adversary knows that, Anders has a low salary between 3k and 5K, he can easily say that Anders has a stomach related disease. The reason for this is l-diversity only considers the diversity of sensitive values within an equivalence group, not their semantic closeness of the values within the group [22]. This is known as similarity attacks. Secondly, there can be another type of attacks that are not addressed by l-diversity, which is known as skewness attacks.

Zip Code	**Age**	**Salary**	**Disease**
476**	2*	3K	Negative
476**	2*	4K	Negative
476**	2*	5K	Negative
476**	2*	6K	Negative
4790*	>40	7K	Negative
4790*	>40	8K	Positive
4790*	>40	9K	Positive
4790*	>40	10K	Negative
476**	3*	11K	Positive
476**	3*	12K	Positive
476**	3*	13K	Positive
476**	3*	14K	Negative
***	***	***	***
488**	>60	16K	Negative

Table 2.6: 10000 Records of a Virus that affects only 1% of the Population

The above table shows 10000 records of a virus infection which affects only 1% of the population. If we look at the first equivalence class in the above table, privacy breaches can be not too impactful, as if someone does not have the disease, they may no care

whether they are revealed or not. The second equivalence class has a 50% chance of having the disease, which is much higher than the original distribution, and the third class has even higher chance. This is called skewness attack, and this happens because l-diversity undertakes the adversaries not to have idea about the distribution of the sensitive attributes, but they can learn it by the table itself. For instance, if an adversary suspects that Tom has the disease, knowing that only 1% of the population has the disease, the adversary believes that Tom belongs to this 1%. So, if he has knowledge on the age and zip code of Tom and he found out Tom belongs in the equivalence class 3, he can easily say that Tom has the disease, looking at the distribution of the equivalence class 3.

t-closeness is proposed as a solution to overcome the above issues by bi Li et al. [27]. An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t. A table is said to have t-closeness if all equivalence classes have t-closeness [27]. Distance is measured using different ways including variational distance and Kullback-Leibler (KL) distance.

The above discussed privacy definitions are widely used and still a lot of research is going on to find improvements for them.

Privacy Model	Description	Advantages	Disadvantages
k-anonymity	A record should be distinguishable from at least k-1 other records.	• There are many existing algorithms written on this • Simplicity	• If an adversary has access to a public data source, he can link a record with it and break privacy (background

			knowledge attacks) • If the sensitive attributes have less diversity, then privacy can be compromised (homogeneity attacks)
l-diversity	Every group of tuples that share the same quasi identifier values in the table have at least l well-represented sensitive values	• Solves diversity issues with sensitive attributes • Solves background knowledge attacks	• Does not consider the distribution of the sensitive attributes.
t-closeness	An equivalence class has t-closeness if the gap between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no	• Solves similarity attacks which are related to the skewness of sensitive attributes	• Relationship between quasi identifiers and sensitive attributes are lost as t increases [5]

	more than a threshold t [5]		

Table 2.7: Summary of Privacy Models

2.3 Real world applications of PPDP

Privacy preserving data publishing and data mining as a whole is used in many different domains and application areas. Out of them, the following areas have obtained a huge attention in the recent past [4].

2.3.1 Mobile data

With the increasing use of mobile devices and ease of installing applications of mobile devices, personal data can be exposed to outsiders in vast amounts. Different analysis can be carried out on mobile data. For instance, text message analysis is something used by WhatsApp in their mobile number verification method [4]. Pervasive techniques involved with mobile devices such as Global Positioning Systems (GPS), can reveal many sensitive information about users [4]. This information can be used to discover individuals' houses, workplaces and the places they visit. Therefore, leakage of location information related to individuals can cause severe privacy violations.

2.3.2 Health care data

Electronic Health Records being a digitization enabler in the health care domain, introduces a risk of privacy disclosure of individuals at the same time. Health records are treated as enormously confidential due to the sensitivity of those data. An emerging field in health care analysis is known as genome sequencing or genome analysis. This is about analyzing genetic sequences of individuals to understand about their DNA patterns. As genetic data can be a very good source which can reveal traits and information about individuals, this is an area of importance which is highly in need for privacy.

Huang [28] introduces a novel algorithm for genome read mapping problem in a privacy preserving way. They claim that the novelty in their algorithm lies in how they

outsource most of the read mapping workload to a scalable public cloud in a privacy preserving way. And they claim that their algorithm outperforms most of the existing genome sequencing algorithms.

Uhlerop et al. [29] bring up the idea of using differential privacy to preserve individual identity when releasing aggregated genomic data. Additive noise is used to preserve privacy before releasing data and they claim that both privacy and utility is preserved in this way.

Gardner et al. [30] argue that most of the existing privacy preservation mechanisms support structured data and they are proposing a mechanism to preserver privacy of both structured and unstructured data in the medical domain through their research.

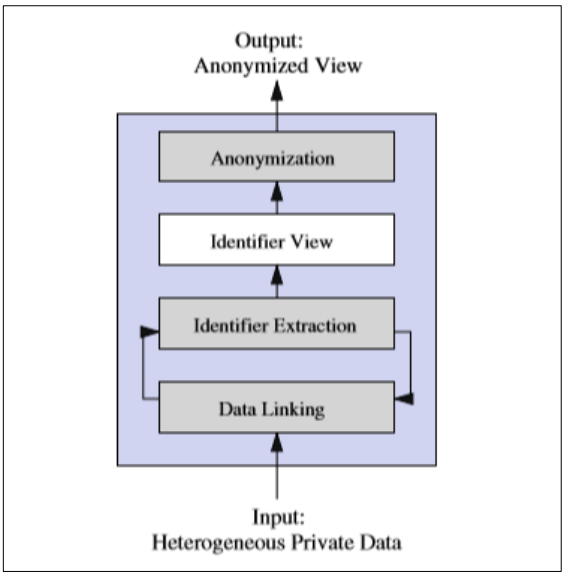


Figure 2.1: Overall Architecture Proposed by Gardner et al. [30]

They have divided the entire process into three steps: identifier extraction, data linking and anonymization. For the identifier extraction task, they have experimented with different types of classifiers. Conditional random field-based classification and prioritized classification with cost proportionate sampling are two such examples. Then they had to link the attributes with individual entities using existing record linkage mechanisms and then perform anonymization on top of linked records. An anonymization model was built on top of k-anonymity and l-diversity using suppression as the technique.

2.3.3 Social media data

Social media being one of the biggest booms in the past decade, lots of personal data are shared across the Internet. Even though people do not think twice while sharing their personal data in social media like Facebook and Instagram, the consequences of any privacy breach can be pretty high. As social media create a huge pool of data with massive variety, this is a big opportunity for anyone who is willing to take the analytical advantage. Facebook status analysis and Twitter tweet analysis are some of the common analysis carried out using social media data.

Various research has been carried out over time to ensure the privacy preservation in social media data. This is relatively challenging that other domains due to the volume and complexity of personal information that involves relationship graphs. Liu et al. [31] conduct a survey on privacy preserving data analysis on social graphs and networks emphasizing the fact that prevailing mechanisms to perform the above is still in its infancy. They bring up the following facts which make privacy preservation on social graphs and networks challenging than normal structured tabular data.

- Modeling background knowledge and the capability of the attacker is comparatively difficult with social networks and graphs. If two nodes in a graph/network are indistinguishable based on some structural metrics, it cannot be assured that those two nodes are indistinguishable based on another set of metrics.
- Quantifying information loss is difficult with graphs.
- Balancing the data utility and information privacy is difficult with graph structures.

Identity disclosure and link disclosure are identified as two problems associated with privacy preservation in social media graphs and the above survey summarizes different research carried out to solve about problems in the social media context [27].

Vadisala et al. [32] summarize challenges in privacy preservation in social networks in a very comprehensive manner.

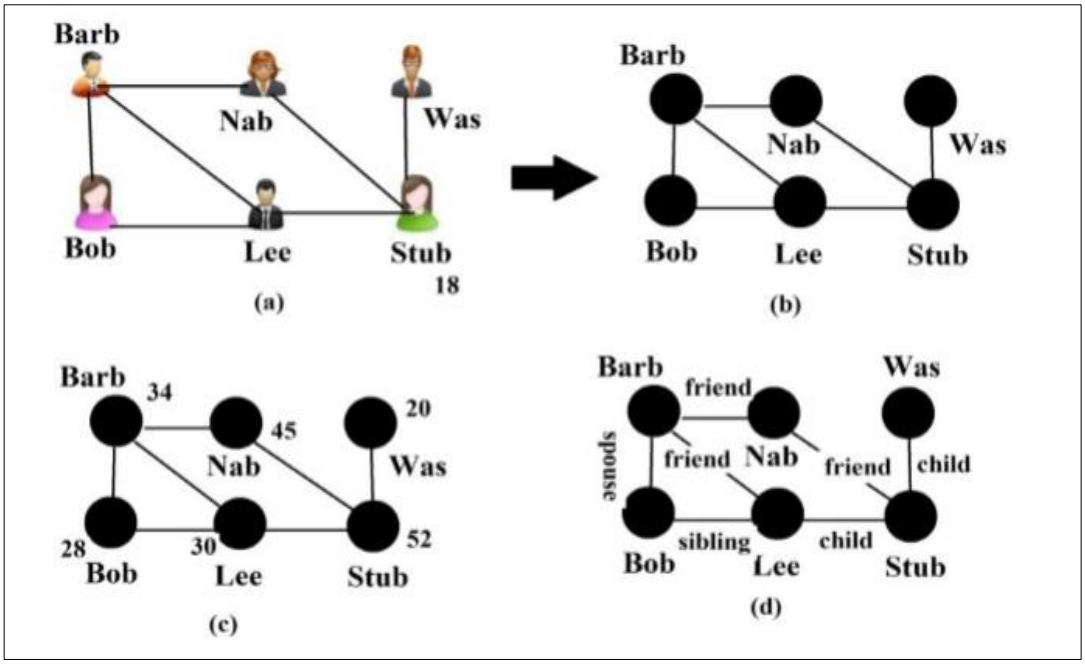


Figure 2.2: Graphical Representation of Social Network Data

In the above survey, the authors categorize privacy breaches into three categories.

- Identity disclosure: Identifying an individual user associated with a node in a social network.
- Link disclosure: Identifying sensitive relationships between nodes in a social network.
- Content disclosure: Identifying sensitive data associated with nodes in a social network.

Further, they summarize different graph anonymization techniques in their survey. Graph randomization, degree-based anonymization, neighborhood-based anonymization, and subgraph-based anonymization techniques are discussed as graph modification techniques. Then, vertex clustering, edge clustering, vertex and edge clustering are discussed as generalization-based anonymization techniques for graphs. Node differential privacy and edge differential privacy are introduced as differential models for privacy preservation in social media graph networks.

Liu et al. [33] come up with a system called LinkMirage which obfuscates social relationships in a social media graph, mainly focusing on link privacy. In their

perturbation-based anonymization mechanism, they use a dynamic clustering approach to find community structures in the graph and then apply selective perturbation in a way that will ensure maximum utility and privacy.

2.4 Privacy metrices

This kind of metrices show how secure the data is from various types pf privacy breaches and adversary attacks. The intention of any privacy preserving mechanism should be to maintain a good balance between the correct level of privacy and utility. These privacy level metrics can be categorized into data privacy metrics and results privacy metrics [5]. Data privacy metrics try to evaluate the ability to infer the original data back from the sanitized data whereas results privacy metrics evaluate how the data mining task can disclose information about original data. Following are some of the privacy metrics belonging to the above two categories that can be found in the literature.

2.4.1 Confidence level

This is one of the earliest data privacy metrices, which is mainly applied in randomization techniques, to evaluate the ability to infer original data from a randomized dataset. If an original value may be estimated to lie in an interval [x1, x2] with c% confidence, then the interval (x2−x1) is the amount of privacy at c% confidence [5]. There is an issue with this specific metric, that it does not consider the initial distribution.

2.4.2 Average conditional entropy

This addresses the issues exist in confidence level metric like not considering about the overall distribution. Agarwal et al. discuss about this metric in their paper about quantifying privacy and the definition given based on information entropy. Given two random variables X and Z the average conditional privacy of X, given Z is $H(X|Z) = 2^{h(X|Z)}$, where h(X|Z) is the conditional differential entropy of X [5].

2.4.3 Hidden failure

This is one of the result metrics which measures the privacy after applying any data mining algorithm. Bertino et al. mention in their survey on privacy quantification techniques that, hidden failure is used to measure the balance between privacy and knowledge discovery [5]. It is defined as the ratio between sensitive patterns hidden with privacy preserving technique and sensitive patterns already existed in original data. In contrast to data privacy metrics, this metric does not measure the amount of information lost.

2.5 Utility metrices

Knowing how useful data is after preserving privacy is equally important as knowing to which extent privacy is protected. Many utility metrices are provided and discussed over time in the literature. And a couple of those metrices are discussed in this section.

2.5.1 Generalization/Suppression counting

A very spontaneous method which existed from the early days of privacy preservation. As the term implies, this is about counting how many generalization/suppression operations were performed. Mayerson et al. [34] used a variation of this to count the number of attributes those were suppressed or generalized. There is a problem with this approach, that the consequence of each generalization operation can be different. For instance, generalizing the value ‘male’ to ‘*’ totally removes the gender data, whereas generalizing the age value 25 to the range 20-30 keeps some information.

2.5.2 Loss metric (LM)

This comes with a solution to the above problem introduced by simply counting generalization operations. This is introduced by Iyengar [35] and Loss Metric is defined as the normalized loss of each attribute of every tuple. The LM for the entire data set is defined as the sum of losses for each attribute.

2.5.3 Discernibility metric (DM)

This designates a penalty to each record on the fact that how many other records in the database are undifferentiated from it, and therefore it works closely in a natural manner under the k-anonymity framework. This metric was proposed by Bayardo et al. [36].

2.5.4 KL divergence

This is a utility metric commonly used in the statistics community and can be used to measure the information loss of sanitized data. Here the original table and the sanitized data are converted to two probability distributions p1 and p2 and the KL divergence is calculated using these two distributions. The larger the value of KL divergence, the greater the information loss [5].

2.5.5 Bivariate measures

For a pair of attributes, A and B, the $\chi 2$ statistic is computed in both the original data and the sanitized data. The $\chi 2$ statistic is then used to compute either the Cramer's V or Pearson's contingency coefficient C. The information loss measure is then considered to be the difference in Cramer's V (or Pearson's contingency coefficient C) from the original data and sanitized data [20].

2.5.6 Workload – aware metrics

LeFevre et al. [37] argue that the utility metrics should depend on the envisioned uses of the sanitized data. This approach suggests that the metric used should be different based on the usage such as classification, regression, or count queries. If the usage is classification, then the corresponding metric should be weighted average of the entropy of the class attribute in each group. If this is a regression problem, then the utility metrics used can be the weighted average of the variance of the class attribute in each group.

2.6 Future directions

Personalized privacy will be a trending direction in the future, mostly due to the fact that the definition of privacy can be subjective from person to person. This implies that giving the user a chance to define to what extent they need the privacy to be ensured. Even though personalized privacy preservation is comparatively a new term, there are a few research works carried out under this area. Xiao et al. [38] define a new personalized privacy model called, personalized anonymity and build a new generalization framework on top of that. They claim that they are going beyond a universal privacy preservation approach and trying to consider individual requirements. Kumar et al. [39] present another approach to achieve personalized privacy preservation using game theory.

Another thought-provoking area that has gained the attention of many scholars is context-aware privacy and this is becoming popular with ubiquitous computing. As the term implies, context-aware privacy is about identifying the privacy requirements based on users' context with the aid of arising Internet of Things (IoT) concepts. As the context can be changed in external factors, defining the privacy based on the context can be a very difficult task. Chakraborthy et al. [40] present a framework called ipShield, which is a framework to enforce context-aware privacy. They point out that, smart phones or mobile devices make privacy breaches easy with the ease of installing third party applications. The framework they suggest, monitors sensors of mobile devices accessed by these untrustworthy applications and provides a risk assessment at the end. Then based on this risk assessment, the users can configure context aware privacy rules.

2.7 Challenges with unstructured data

Privacy preservation has been challenging specially with semi structured and unstructured data, due to different qualities of these data. Even though unstructured data can hold a huge amount of beneficial information to conduct analysis, certain attributes of these data can make their usage a little complicated. Data we gather from social media posts, call center conversations, emails and Tweets can be tagged as

semi/unstructured data as they do not fit into a structured database model. A couple of such challenges associated with unstructured data, especially in the privacy preserving context is discussed below.

2.7.1 Difficulty in identifying sensitive attributes

Due to the word-based and unstructured nature of these data, extracting sensitive attributes is not easy or direct. There should be some special mechanisms to identify and tag attributes in order to anonymize or sanitize them.

2.7.2 Volume of data

Knowingly or unknowingly many platforms are collecting data of massive volumes. Sometimes authoritative bodies are not capable of handling that much data volumes, so they tend to lack the awareness of data they have in hand. This might lead to severe privacy breaches. And this attribute of unstructured data itself can make privacy preservation hard specially when extracting sensitive attributes precisely.

2.7.3 Quality of data

Quality of data can be low with unstructured data. For instance, people tend to post false details about themselves in social media or in other places as a result of growing privacy concerns. For example, they can say a completely false marital status [41], and this can impact on the utility on any analysis carried out on these data. Privacy preserving won't experience its true utility, when data itself is distorted with external factors.

2.8 Unstructured privacy preserving data publishing scenarios

There can be different use cases which require anonymization of unstructured data. These data might belong to different domains. Vincze et al. discuss about a few different domains which can contain unstructured sensitive data [42].

Medical records hold an importance among these scenarios because they can contain really crucial sensitive information. Clinical documents which contain personal health

information can include textual data which cannot be exposed publicly. Patient names, health proxies, family member details, doctor details, phone numbers, hospital names and geographic locations are a few such privacy related attributes in the medical domain.

They mention social media data as one of the most prominent platforms, which contains a wealth of privacy related attributes in its contents. For example, social media contents like Facebook posts and tweets can be used in personality classification and micro-targeted disinformation campaigns.

Then they introduce another interesting source of information, that can contain textual privacy related attributes. That is employee curriculum vita or CVs. CVs can also reveal a lot about individuals as they contain information like personal data, education, competencies, and hobbies etc.

2.9 Unstructured data anonymizing techniques

There are a couple of research works in the literature, which has attempted to anonymize unstructured data. Sweeny [43] proposes a mechanism to pseudo anonymize textual medical data via named entity recognition. This includes simple named entities such as location, country, names, and direct replacing them with a pseudo value.

Neamatullah et al. proposes another mechanism which is based on anonymization of free-text medical records based on lexicon-based and context -related checks. And they simply replace the values with a categorical representation [44]. Mottwani et al. propose a mechanism to anonymize textual data in a way which individuals are explicitly characterized by their distinctive properties [45]. The United Kingdom Data Archive (UK DA) utilizes a mechanism where numbers and letters starting with a capital latter are simply replaced [46]. Vico et al. have gone for a named entity-based anonymization technique, where they have pipelined a set of existing natural language processing tools and simply replaced the extracted named entities [47]. Kleinberg et al. discuss another approach where they have gone for a named entity plus a rule-based

anonymization approach [48]. There rules were mapped to certain word types and based on that words were anonymized.

2.10 Summary

There are various privacy preserving techniques available in the literature as well there a number of predefined privacy models as well, which act as industry norms of performing privacy preservation. And also, there are different privacy and utility metrics which can be used to evaluate the anonymization operations. Personalized privacy and context aware privacy are some future directions. When it comes to unstructured data, there are a set of inherent challenges and there are some research work which was carried out on this subject.

CHAPTER 3: METHODOLOGY

This chapter discusses about the proposed solution and approach to implement the suggested privacy preserving data publishing middleware. First, high level architecture of the system will be discussed and then each component of the proposed solution will be discussed in detail. These details will include design of individual components, algorithms used, and metrics incorporated for evaluation purposes. When it comes to metrics, both privacy and utility metrics will be discussed.

3.1 High-level architecture

The objective of this research was to come up with an end to end system which can realize the concept of privacy preserving data publishing in an unstructured context. The suggested proof of concept solution will act as a middleware that can be used by any wrapper application to publish Twitter data to a source of interest for analytical purposes. Even though the proof of concept is built with Twitter as the base, the system is implemented in a way where it can be generalized for any other platform of any other source of structured data in the future. In order to support the potential for future extensions, the core parts of the PPDP workflow are implemented as Restful API endpoints. Therefore, the PPDP workflow and the utilizing system can be completely decoupled from each other. And also, components within the workflow itself are loosely coupled from each other and separated into different end points that can be utilized from a data publisher.

With the intention of letting the user try out and understand the PPDP workflow, a sample client application too was implemented as a method of demonstrating PPDP workflow utilization. This client application which is a web application is capable of simulating a data publishing task and performing the sanitization on that published dataset using the proposed middleware. Other than that, the sample client has the ability to visually inspect the utility metrics computed for the privacy preserving data publishing task it performs. As the test data generation methodology, a live search is performed on Twitter using some keywords with the aid of Twitter's public API and then a data set is created.

Both of this client application and the Restful API compose the overall system. The Restful API, which is the core of the proposed system comprises of following functionalities.

- Accepting textual data set as input and reconstruct the data set and store in a csv file after applying the privacy preserving data publishing mechanisms.
- Extracting personal privacy related attributes from textual data. These attributes can be direct identifiers or quasi identifiers which was discussed in the previous parts of the report. Other than those, sensitive attributes, that people don't want to reveal will be extracted too. Some machine learning models are employed to extract personal information and textual data will be automatically tagged with respective tags to indicate privacy attributes. This is the most crucial part of the application as that is the component which enabled privacy preservation of unstructured data, as attribute extraction is the most difficult task there. Further details will be discussed in the latter parts of this chapter.
- Next functionality is the anonymization of extracted privacy attributes. The API caters towards the anonymization of the supported direct and quasi identifiers using some sanitization mechanisms in the literature. Methods like generalization, union, suppression and swapping are used to handle the anonymization. The most suitable anonymization mechanism will be applied based on the nature of the attribute.
- Then the API provides endpoint to evaluate the utility of the anonymized data set. Utility metrics like generalization counting, loss metric and discernibility metric are used from the literature to achieve this task.

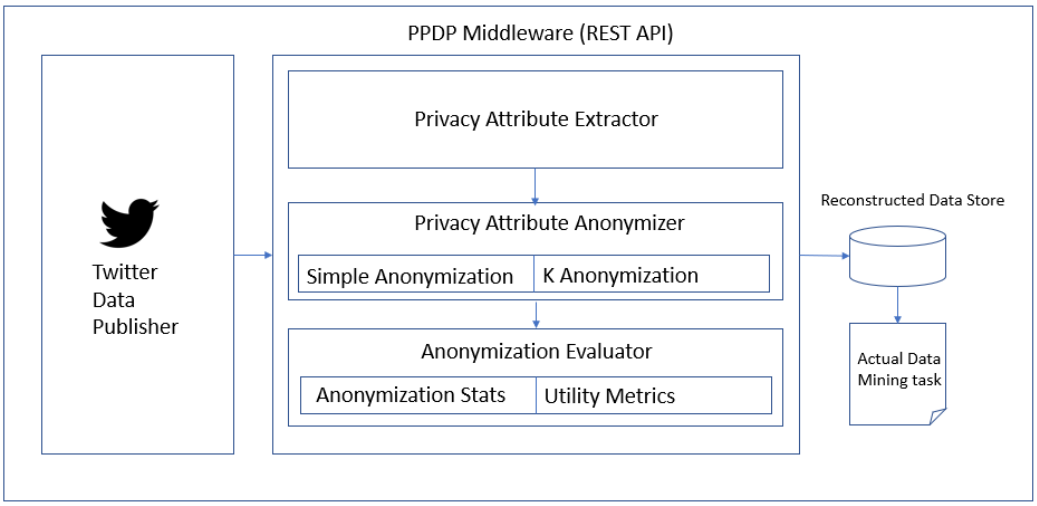


Figure 3.1: High Level Architecture

3.2 Technologies adopted

As depicted in figure 3.1 the overall system comprises of 2 main parts: The Restful API and the sample client application or the Twitter Data Publisher.

The Restful API was implemented suing Python 3 and Flask web framework. The endpoints were written adhering to rest principles in a way which can be consumed by a variety of rest clients. Model building and model evaluation was also done using Python and its libraries. Different libraries utilized for various tasks of the implementation will be discussed when describing each component separately. The API is well documented using Swagger and the figure 3.2 shows a snapshot of the Swagger API documentation. This well documented API enables any data publisher to try and understand the PPDP workflow tasks separately before starting with their actual data publishing mechanism.

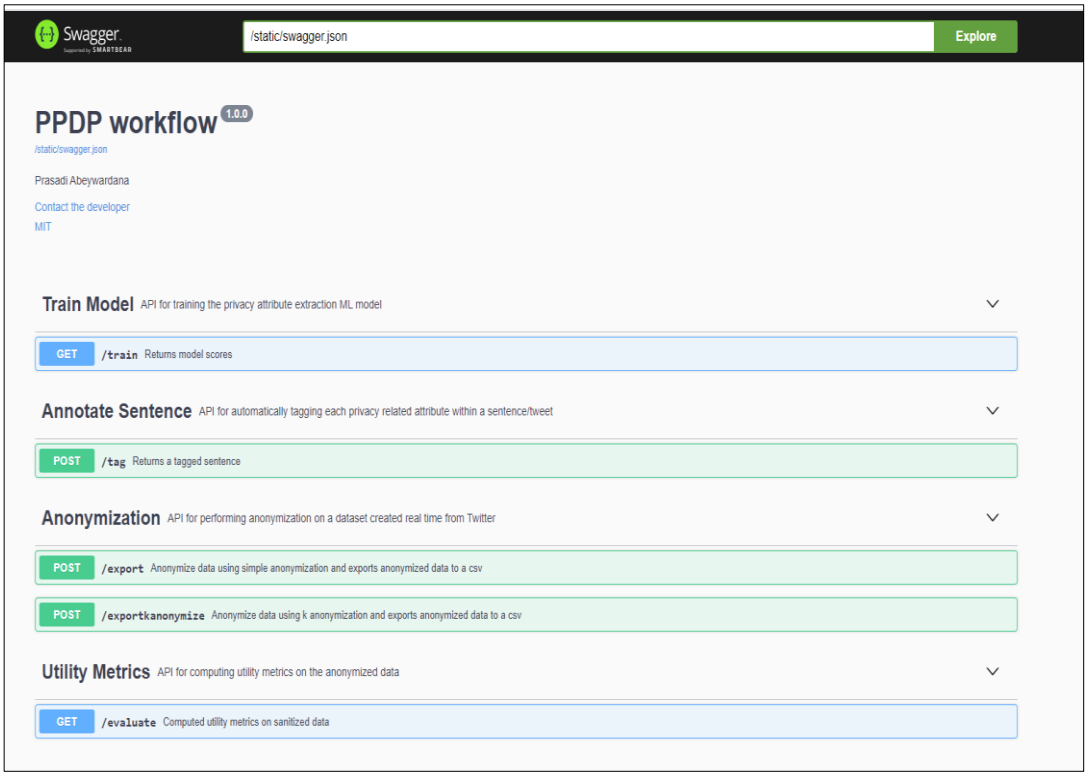


Figure 3.2: API Documentation

The sample web client was implemented as an Angular application and it consumes the above-mentioned Restful API. Google charts were used to visualize the metrics in an interactive way. Each component of the middleware will be explained in detail next.

3.3 Privacy attribute extractor

This is the most critical component of the entire middleware as this module is responsible for extracting privacy attributes from unstructured data, which is the most challenging task with the unstructured nature of the data. This module initiates the PPDP workflow by extracting any attribute from textual data that can disclose individual information. This component adopts various machine learning and natural language-based processing in extracting the privacy attributes.

3.3.1 Twitter corpus

One of the main contributions of this research was to come up with a Twitter corpus, where words were tagged with privacy attributes. There was no publicly available

dataset which targets unstructured data tagged with privacy attributes, specifically social media data accessible for the research community. Therefore, as a part of this research, a comprehensive Twitter corpus was built and annotated with privacy related attributes. Annotations were done manually by 3 annotators adhering to a predefined annotation schema. This annotation schema consisted of direct identifiers, quasi identifiers and sensitive attributes.

Attribute Type	**Tag**	**Attribute**
Direct Identifiers	DI	Name, TwitterId
Quasi Identifiers	QIAGE	Age
Quasi Identifiers	QIRELIGION	Religion
Quasi Identifiers	QIREGION	Region
Quasi Identifiers	QIGENDER	Gender
Quasi Identifiers	QILANG	Language
Quasi Identifiers	QIJOB	Occupation
Quasi Identifiers	QIMARITAL	Marital Status
Sensitive Attribute	SA	Health Conditions, Relationship Status, Salary
Non-Sensitive	NONE	Anything that does not belong to above

Table 3.1: Annotation Scheme

Table 3.1 depicts the annotation scheme used in the manual annotation process. Quasi identifiers used in the annotation process were carefully selected from the literature. Quasi identifiers used here are the most common quasi identifiers used in the context of social media and in general. And also, when picking the sensitive attributes, a couple of commonly known attributes such as health conditions, relationship status, salary and political preference were used. Table 3.2 shows some statistics computed on the corpus.

Parameter	**Value**
Tweets	3000

Sentences	4980
Words	62649
Words per tweet	21
Words per sentence	13
Sentences per tweet	2

Table 3.2: Corpus Statistics

Additionally, table 3.3 shows annotation statistics which includes counts for each tag category. This data can give an understanding of distribution of tags in the dataset. Looking at the values of the table 3.3, it is clear that the corpus contains a well-balanced set of data that can be used in any PPDP related machine learning task in the future.

Tag	Word Count	Tweet Count
DI	2496	2178
QIAGE	156	120
QIRELIGION	145	112
QIREGION	306	281
QIGENDER	710	524
QILANG	164	106
QIJOB	275	199
QIMARITAL	113	71
SA	1563	1079

Table 3.3: Annotation Statistics

The tweets for the corpus was selectively picked from a public Kaggle dataset which was intended for a sentiment analysis task [57]. This initial dataset contained 1.6 million tweets, so it contained a variety of tweets which supports the intention of this

research. A list of keywords in each category of tags were prepared and a manual search was performed on the above-mentioned Twitter corpus to extract the tweets of interest. In that way a corpus of 3000 tweets were built and then manually annotated by three annotators with privacy attributes.

3.3.1.1 Inter rater agreement

Absolute agreement was calculated per tag type considering all the three annotators. Additionally, Cohen's kappa statistic $((P_o-P_e)/(1-P_e))$ was calculated by considering each pair of annotators and averaged between pairs per tag category. The results are shown in Table 3.4.

Tag	**Agreed Tag Count**	**Absolute Agreement Percentage** **(Agreed Tag Count by All Annotators/Total Tag Count)**	**Kappa Coefficient**
DI	2490	99%	0.92
QIAGE	156	100%	0.97
QIRELIGION	130	89%	0.70
QIREGION	288	94%	0.72
QIGENDER	603	84%	0.68
QILANG	132	80%	0.65
QIJOB	254	92%	0.73
QIMARITAL	85	75%	0.59
SA	1115	71%	0.54

Table 3.4: Inter-rater agreement results

It can be seen that the absolute agreement was above 70% for all the tag categories. And the kappa value is above 0.5 for all the tag categories. This demonstrates a good inter-rater agreement. Agreement for sensitive attributes seems to be the lowest, mostly due to its subjective nature and agreement for age is at the highest as it is something straightforward.

3.3.2 Data preprocessing

A set of utility methods were employed to do some initial preprocessing to tweets. Tweets include some urls and removing them was the first step of preprocessing. Once that was done, rest of the special characters were removed from tweets too. Case of the words were kept as it is without doing any transformation, because they were used as features at a later stage.

3.3.3 Data transformation

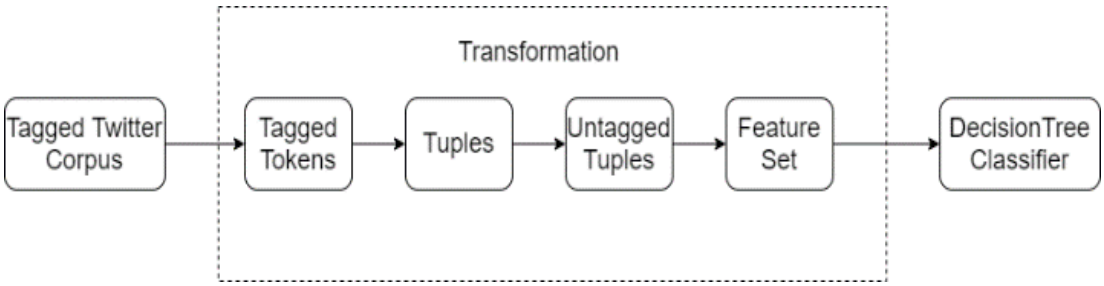


Figure 3.3: Tweets Transformation Process

The preprocessed corpus was transformed into a set of features before applying any machine learning algorithm. A set of utility methods were implemented in order to perform the transformation process. Figure 3.3 depicts the transformation workflow used in converting the corpus into a set of features. It is important to understand that features were computed per tweet/post, so the transformation process was repeated for every tweet in the corpus.

Corpus was separated into a feature set and a label set before being fed into the classifier and what is depicted in Figure 2 is the process of separating the feature set. First the tweets were tokenized using the python nltk tokenizer. Then the tagged tokens were converted into the tuple form using the same library. The next step in the feature extraction process was to untag the tuples, so that the words can be used independent of the tags to obtain features. In order to perform the untagging, a utility method was used. And then the next step was actual feature extraction part and this step was repeated for each word in a tweet. This was also achieved through a custom-made utility method. In order to get an idea about the features to be used a research work done by Carreras et al. was referred [49]. This research was concentrated around a

named entity extractor which was implemented using AdaBoost. They precisely explain their feature selection approach and referring to that research was really helpful in understanding the feature types to be used, because the intention of the features was somewhat similar in both the research works.

The entire feature representation used in the transformation process can be categorized in the following manner.

- Orthographic Features: This means properties of words related to how it is capitalized (initial-caps, all-caps), the kind of characters that create the word (contains-digits, all-digits, alphanumeric, roman-number), the presence of punctuation marks (contains-dots, contains hyphen, acronym), single character patterns (lonely initial, punctuation-mark, single-char), or pattern (URL) [49].
- Affix Features: This indicates the prefixes and suffixes of the word. Specifically, in the above transformation, prefixes and suffixes were computed up to 3 characters. Based on the character count, prefixes and suffixes were treated as separate features.
- Syntactic Features: This basically represents the part-of-speech tags of words. POS tags were computed and appended as a feature of the word itself here.
- Gazetteer Features: This implies the classes obtained by an external gazetteer used to determine possible classes for each word. Spacy [50] was used as the source of external gazetteer here and it will be discussed with further information in this chapter.

The method used to extract gazetteer features should be highlighted here. An external named entity recognizer called spaCy [50] was used in identifying the gazetteer features. spaCy includes an enormously rapid entity detection system, that allocates labels to adjacent areas of tokens. The defaulting model discovers a range of numeric and named entities, involving companies, locations, organizations, and products. Extracted named entity labels from this spaCy tool was utilized as features in the transformation process. Table 3.6 shows the named entities recognized through spaCy.

The full list of features is shown in table 3.5.

Feature	Feature Type
has_hyphen	Orthographic
is_numeric	Orthographic
is_capitalized	Orthographic
is_all_caps	Orthographic
is_all_lower	Orthographic
prefix-1	Affix
prefix-2	Affix
prefix-3	Affix
suffix-1	Affix
suffix-2	Affix
suffix-3	Affix
prev_word	Orthographic
next_word	Orthographic
is_first	Orthographic
is_last	Orthographic
pos_tag	Syntactic
named_entity	Gazetteer

Table 3.5: Feature List

PERSON	People, including fictional.
NORP	Nationalities or religious or political groups.
FAC	Buildings, airports, highways, bridges, etc.
ORG	Companies, agencies, institutions, etc.
GPE	Countries, cities, states.
LOC	Non-GPE locations, mountain ranges, bodies of water.
PRODUCT	Objects, vehicles, foods, etc. (Not services.)
EVENT	Named hurricanes, battles, wars, sports events, etc.
WORK_OF_ART	Titles of books, songs, etc.
LAW	Named documents made into laws.

LANGUAGE	Any named language.
DATE	Absolute or relative dates or periods.
TIME	Times smaller than a day.
PERCENT	Percentage, including "%"
MONEY	Monetary values, including unit.
QUANTITY	Measurements, as of weight or distance.
ORDINAL	"first", "second", etc.
CARDINAL	Numerals that do not fall under another type.

Table 3.6: spaCy Named Entities [50]

It is evident through the above table that spaCy itself contains a wide range of named entities and inputting them as features can be really helpful in the automatic annotation process.

Once the features are extracted, they were used in the model training process. A classification-based approach was used to tag the tweets automatically. A set of classification models were trained with the data set and their accuracies were compared.

When preparing the training and testing data, n fold cross validation was used with the n value of 5. After the training and testing data sets were separated for each split, tweets were transformed into feature vectors using the previously mentioned transformation mechanism and the classifiers were trained using those data. Table 3.7 shows various classifiers tried out in building the automatic annotator.

Classifier	**Description**	**Parameters**
Decision Tree	Works by cracking down a dataset into smaller divisions based on distinct conditions [51].	Criterion: entropy, Splitter: best
Naïve Bayes	Decides the probability that an instance fits to some class, estimating the probability that an outcome will occur provided that	

	some input incident has already occurred [51].	
K-Nearest Neighbor	Works by examining the distance from some test instance to the recognized values of some training instance. The cluster of data points that would give the tiniest distance among the training points and the testing point is the class that is chosen [51].	n_neighbors: 5
Support Vector Machine	Performs by sketching a line between the different clusters of data points to band them into classes. Points on two sides of the line will belong to two clsses [51].	Kernel: Linear

Table 3.7: Classifiers Attempted

Classification based tagging is a well-known practice in the research literature. Different classifiers like Support Vector Machines and Decision Trees as well as ensemble mechanism like Random Forests have proven well with tagging tasks [51]. Therefore, this research also focused on going for a classification based automatic tagger.

Once the above-mentioned models were trained using the dataset, confusion metrics were computed for each split in the k-fold cross validation and averaged across splits.

Table 3.8 shows the macro average of F1 score and the weighted average for each algorithm. Weighted average stays at a pretty high value, because it is manipulated by the high proportion of the ‘None’ tag.

Algorithm	Macro Average	Weighted Average
Decision Tree	0.6122	0.91
Naïve Bayes	0.5664	0.91
KNN	0.6330	0.93
SVM	0.7245	0.94

Table 3.8: Aggregated Accuracy of the models

The evaluation results show that SVM has the best accuracy out of the classifiers tried, therefore it is used for the workflow steps from tagging onwards.

Following is a sample tweet automatically tagged through the tagging model.

('My', 'None')('teacher', 'QIJOB')('who', 'None')('lived', 'None')('in', 'None')('USA', 'QIREGION')('died', 'None')('of', 'None')('cancer', 'SA')('at', 'None')('age', 'None')('65', 'QIAGE')

All the new tweets published through the data publisher will go through this module and get automatically tagged with appropriate tags.

Table 3.9 lists all the python libraries and their objectives utilized in implementing the automatic tagger/privacy attribute extractor.

Library	**Purpose**
spacy	Named entity extraction
pandas	Data structure manipulation
nltk	Corpus transformation
sklearn.feature_extraction	Feature vectorization
sklearn.pipeline	Model creation
sklearn.tree	Decision tree algorithm
sklearn.naive_bayes	Naïve Bayes algorithm

sklearn.neighbors	KNN algorithm
sklearn.svm	SVM algorithm
sklearn.model_selection	k-fold cross validation
sklearn.metrics	Model evaluation
joblib	Saving the models

Table 3.9: Libraries used for Implementing Automatic Tagger

3.4 Privacy attribute anonymizer

After extracting privacy attributes through the classification based automatic tagger, the next step of the PPDP flow is to anonymize the extracted attributes. The prototype supports two major ways of performing the anonymization.

1) Simple anonymization: This method takes a tweet as an input and perform anonymization simply based on the deidentification mechanism allotted for each privacy attribute type. Therefore, each direct identifier and quasi identifier in the tweet will be anonymized. This anonymization mechanism does not take into account, other tweets in the dataset or relationship between privacy attributes within a tweet itself, but it blindly sanitizes data in a way privacy attributes will not be disclosed.

2) K-Anonymization: This performs anonymization based on the well-known privacy model called k-anonymity model. In order to perform this, first the data is converted into structured format. Then k-anonymity model is applied on this structured dataset and anonymization is performed. This takes into account other tweets in the dataset as well as treats the privacy attributes within a tweet as a single entity. A set of tweets with the same quasi identifiers is said to be belonged to the same equivalence class. This methodology performs anonymization based on the concept of equivalence classes.

These two anonymization mechanisms will be described in detail within this section.

3.4.1 Simple anonymization

This mechanism simply goes through each direct identifier and quasi identifier of the tweets and applies a predefined anonymization technique based on the type of the identifier. Table 3.10 shows the anonymization scheme defined for privacy attributes.

Attribute	Sanitization Technique	Original Value	Sanitized Value
Name, TwitterId	Complete Anonymization	John	******
Age	Generalization (to a number range)	65	60-70
Religion	Generalization	Hindu	<Religion>
Region	Generalization	Sri Lanka	<Region>
Gender	Generalization	Female	<Gender>
Language	Generalization	English	<Language>
Occupation	Generalization	Teacher	<Job>
Marital Status	Generalization	Married	<Any>

Table 3.10: Anonymization Scheme

When applying anonymization under the above schema, the fact that how distinguishable a certain set of attributes compared to other tweets in the data set is not considered. So, it blindly applies the matching anonymization technique from the above table when the matching direct or quasi identifier type found. This mechanism will be useful for data publishing with thorough privacy requirements and less focus on the quality of the analysis.

3.4.2 K anonymization

This is a privacy model which tries to protect individual privacy by grouping data into clusters known as equivalence classes. This model is discussed in detail in the literature review chapter. This method assumes that we have a data set of N entries and each entry consists of a list of D attributes where some of these attributes are quasi identifiers which can be combined to uniquely identify an individual. In addition to that this model assumes there is one sensitive attribute in the dataset that the person does not want to reveal.

K-anonymity needs that we group individual records of our dataset into group of at least k records and replace the quasi identifier attributes of these records with anonymized values, such that it is no longer possible to identify the individual values. This protects individuals by ensuring that an adversary who knows all values of a person's quasi identifier attributes can only find out which group a person might belong to but not know if the person is really in the dataset.

As it sounds, converting a dataset into a k-anonymous dataset is a complex problem, but there are good enough approaches discussed in the literature. Most of those approaches use a greedy search methodology. This research utilizes an implementation of an algorithm called ‘Mondrian’ algorithms which is intended to solve the above problem.

3.4.2.1 Mondrian algorithm

This uses a greedy search algorithm to partition the original data into smaller groups [53]. The algorithm presumes that we have transferred all attributes into numerical or categorical values, and we are able to measure the "span" of a given data attribute X_i. The partitioning flow used in the implementation is depicted in detail in figure 3.4.

1. Initialize the finished set of partitions to an empty set $P_{finished} = \{\}$
2. Initialize the working set of partitions to a set containing a partition with the entire dataset $P_{working} = \{\{1, 2, \ldots, N\}\}$
3. While there are partitions in the working set, pop one partition from it and
    - Calculate the relative spans of all columns in the partition
    - Sort the resulting columns by their span (in descending order) and iterate over them. For each column,
        - Try to split the partition along that column using the median of the column values as the split point
        - Check if the resulting partitions are valid according to our k-anonymity (and possibly additional) criteria
        - If yes, add the two new partitions to the working set and break out of the loop

Figure 3.4: Data Partitioning Process

After partitioning the dataset, quasi identifier values within each partition must be aggregated or anonymized using a preferable anonymization mechanism. Quasi identifiers are labelled as categorical or numerical in order to determine the type of anonymization to be performed.

Quasi Identifier	**Type**	**Anonymization Technique**
Age	Numerical	Mean of the partition
Religion	Categorical	Union of the partition
Region	Categorical	Union of the partition
Gender	Categorical	Union of the partition
Language	Categorical	Union of the partition
Occupation	Categorical	Union of the partition

Marital Status	Categorical	Union of the partition

Table 3.11: Anonymization Techniques Applied

A utility function was implemented to calculate spans of all columns in a partition. For numerical values max-min was computed and for categorical values number of unique values in the category was computed.

```
def partition_dataset(df, feature_columns, sensitive_column, scale, is_valid):
    """
    :param               df: The dataframe to be partitioned.
    :param  feature_columns: A list of column names along which to partition the dataset.
    :param sensitive_column: The name of the sensitive column (to be passed on to the `is_valid` function)
    :param            scale: The column spans as generated before.
    :param         is_valid: A function that takes a dataframe and a partition and returns True if the par
tition is valid.
    :returns               : A list of valid partitions that cover the entire dataframe.
    """
    finished_partitions = []
    partitions = [df.index]
    while partitions:
        partition = partitions.pop(0)
        spans = get_spans(df[feature_columns], partition, scale)
        for column, span in sorted(spans.items(), key=lambda x:-x[1]):
            lp, rp = split(df, partition, column)
            if not is_valid(df, lp, sensitive_column) or not is_valid(df, rp, sensitive_column):
                continue
            partitions.extend((lp, rp))
            break
        else:
            finished_partitions.append(partition)
    return finished_partitions
```

Figure 3.5: Partitioning function

Then another utility function is implemented as a split function which splits a given partition into two partitions based on a split value. Here also, for numerical columns, the splitting point is median on the selected column and for categorical values the splitting point is number of unique values of the given column.

There is one last utility function that validates whether a given partition adheres to the k-anonymity rule. Using the above helper functions, the data set is repetitively broken into partitions in a way they are adhering to the k-anonymity rule. The value of k is configurable in the implementation. Figure 3.5 shows the complete partitioning method, which utilizes the above described helper functions. This partitioning algorithm runs with O(nlogn) complexity where n = |T| for relation T. The Mondrian algorithm is based on the concept of multidimensional partitioning. According to this, multidimensional regions are recognized first across the domain space and then anonymization is performed in each region using summary statistics. First step of the process is happening in a recursive way.

In order to apply k-anonymization, textual data has to be converted to structured format and persisted as an intermediate step. For this purpose, a MongoDB collection was used.

Figure 3.6: Textual data converted to structured format

Using a MongoDB as the intermediate store handled the complexities of accessing and processing traditional relational databases. And also using a nosql database like this can cater towards high data volume problems, that comes with this kind of data publishing tasks. Figure 3.6 shows some records from the MongoDB which holds quasi identifiers and sensitive attributes that come through textual data. When converting textual data to structured format following assumptions were made.

1. One sentence is equal to one record
2. One sentence talks about only one individual
3. If there are more than one quasi identifier of the same category in a sentence, only the last value will be considered in building the structured dataset

AgeGender	Job	Region	SA	CountRows	
she,25M		,East		4	1,2,3,4
she,25M		,East	Cancer	1	5
she,25M		,East	cancer	3	6,7,8
He,Aunt		Bhilwara...,@BoSnerdleycameos		2	25,26
He,Aunt		Bhilwara...,@BoSnerdleycancer		4	23,45,47,48
He,Aunt		Bhilwara...,@BoSnerdleycauses		1	24
He,man		Omaha..	Cancer	4	34,35,36,37

Figure 3.7: K-Anonymized data frame

Figure 3.7 shows a data frame which was 4-anonymized. If we look into the quasi identifier groups and the row counts, we can see there are at least 4 records in each quasi identifier group that means every record is at least indistinguishable from 3 other records.

Table 3.11 shows the libraries used in data anonymization module.

Library	**Purpose**
pymongo	Store intermediate structured data in MongoDB
pandas	Managing data structures

Table 3.12: Libraries used for the data anonymization module

3.5 Utility evaluator

The prototype allows the user himself to compute some utility metrics provided in the literature on the anonymized dataset. These measures specifically target the quality of the quasi identifier groups. Providing a functionality like this will allow the user to assess the quality of anonymized dataset before publishing data to be used by a third party.

3.5.1 Discernibility metrics

This designates a penalty to each record on the fact that how many other records in the database are undifferentiated from it [5]. If a tuple fits to quasi identifier class of size n, then the penalty for the tuple will be n and the penalty for the class will be n^2. Whenever an anonymization task is performed, user is given the ability to calculate the discernibility metrics for each quasi identifier. The specialty with discernibility metric is it can compare the cost of generalizing for each qid value. Higher the discernibility value, higher the cost of generalization is.

3.5.2 Loss metrics

This calculates the normalized loss of each attribute of every tuple. This, in particular targets the data damage caused by the generalization. Loss Metric is specified as the count of nodes a tuple's value has been made impossible to differentiate from (via generalization) in comparison with the total count of initial leaf nodes in the taxonomy tree [5]. Loss metric is created as n-1/m where n is the number of descendants of a parent value in a generalization tree and m is the total number of domain values of an attribute.

3.5.3 Generalization counting

This counts how many generalization/suppression operations were performed.

Allowing the user to evaluate these metrics by him/herself will give an opportunity to understand to which extent the data quality is preserved. The sample web client showcases these metrics in a graphical manner to demonstrate the usage of these metrics.

Other than that, there is a sample experiment that can be performed in the sample web application, which performs a simple text classification task before and after anonymization and evaluate the results.

3.6 Sample web client

A web client application was implemented to demonstrate how the privacy preserving middleware can be utilized in the real world. This is a simple application written in Angular framework. It gives the user the ability to try out PPDP workflow tasks separately and as a whole. The web application supports following functionalities.

- View automatic tagging model scores/accuracy details
- Perform tagging and anonymization of an individual sentence/tweet
- Perform simple anonymization on a sample dataset provided
    - Download the anonymized dataset
    - View utility metrics
- Perform k-anonymization on a sample dataset provided
    - Download the anonymized dataset
    - View utility metrics
- Perform simple anonymization on a dataset created through Twitter API real time
    - Download the anonymized dataset
    - View utility metrics
- Perform k-anonymization on a dataset created through Twitter API real time
    - Download the anonymized dataset
    - View utility metrics

- Perform an actual classification task on a normal dataset and an anonymized dataset and compare the result accuracy

The main objective of implementing a sample web client application was to motivate the adoption of the framework. Because having a platform to play around and try out things by ourselves will be very helpful when utilizing the framework features in a different domain or a context.

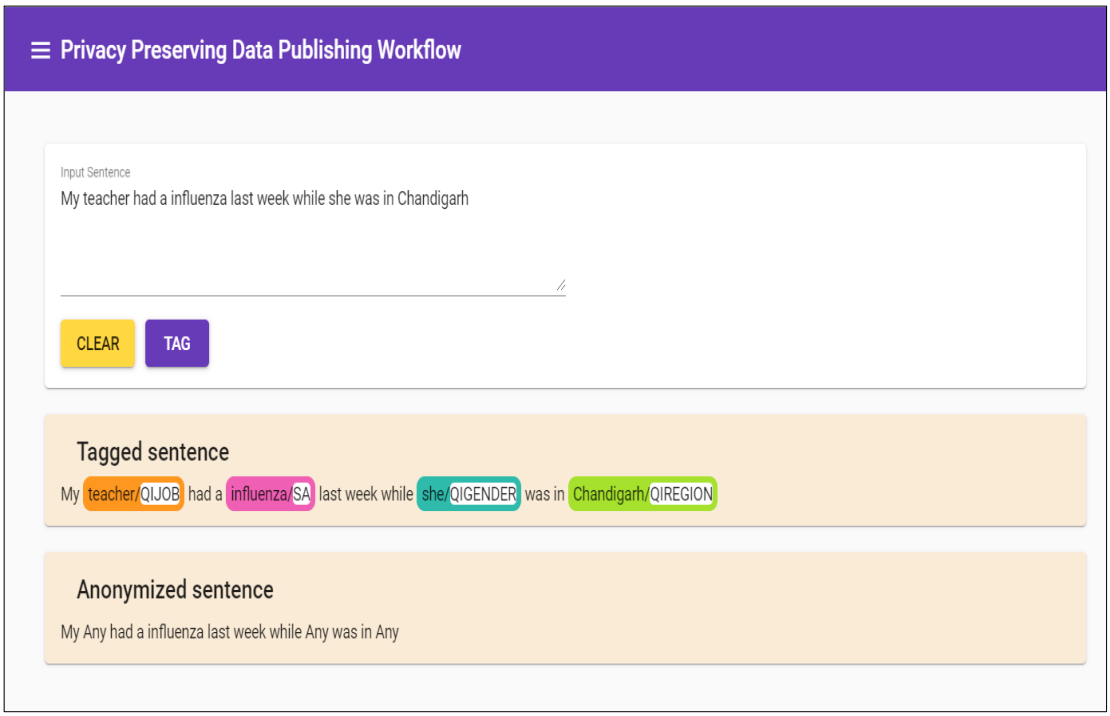


Figure 3.8: Snapshots from the web client

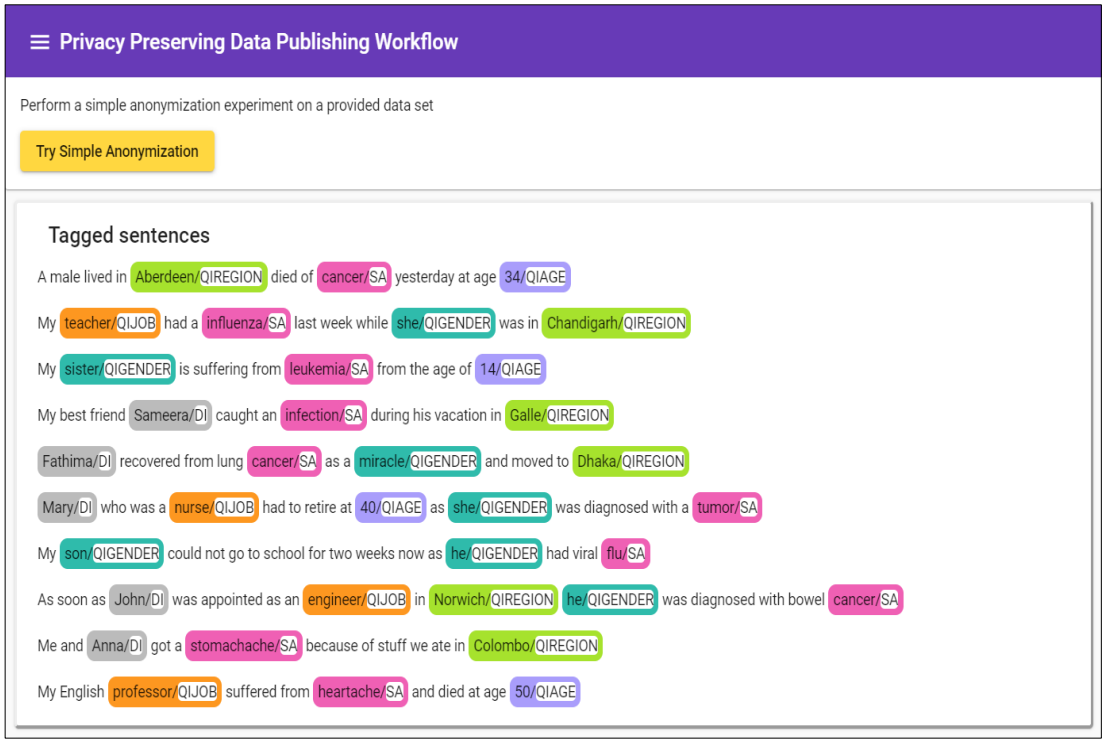


Figure 3.9: Snapshots from the web client

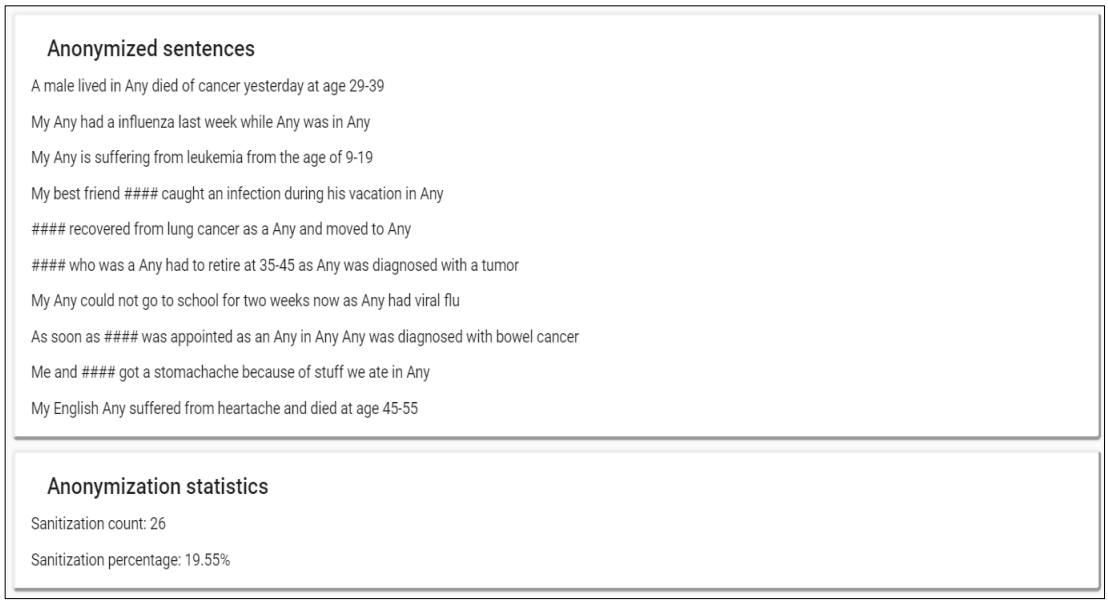
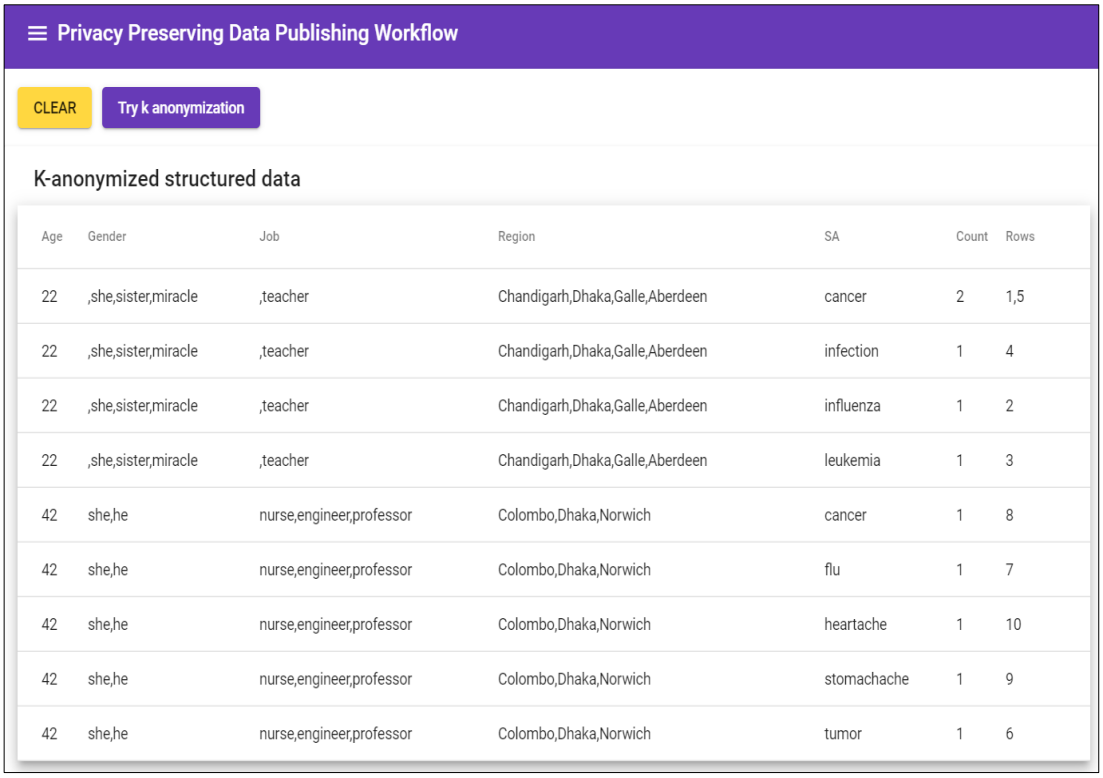


Figure 3.10: Snapshots from the web client



Age	Gender	Job	Region	SA	Count	Rows
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	cancer	2	1,5
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	infection	1	4
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	influenza	1	2
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	leukemia	1	3
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	cancer	1	8
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	flu	1	7
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	heartache	1	10
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	stomachache	1	9
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	tumor	1	6

Figure 3.11: Snapshots from the web client

3.7 Summary

This chapter has discussed the proposed solution in detail, starting with its high-level architecture and then digging deeper into its individual components. In addition to discussing individual components, the technology stack used is also described. Each component is supported with respective algorithm pseudo codes, diagrams and tables to get a thorough understanding of the system design.

CHAPTER 4: EXPERIMENTAL DESIGN AND RESULTS

The main focus of this chapter is to design and discuss the results of experiments that evaluate different components of the prototype designed. Automatic tagger accuracy will be the main parameter that will be evaluated. Additionally, a few more experiments are designed to showcase the usage of a real life PPDP workflow/pipeline. Through these experiments it can be understood, how the proposed system would be utilized.

4.1 Evaluating privacy attribute annotator accuracy

A couple of evaluation techniques were used to validate how well the privacy attribute extracting model is performing on different tag categories. Experiments were designed in a way where a set of benchmark values were generated for future users of the concept.

4.1.1 k-fold cross validation

At the point of building the classification model for automatic tagging, k-fold cross validation was used when splitting training and test data sets. There were a couple of reasons to use k-fold cross validation than a simple 70% - 30% split. Four types of classifiers were tried, and each classifier was trained using k-fold cross validation. Number of folds used per model was 5. So according to this our 3000 tweets were split into 5 sub datasets. Out of these 5 folds, 4 folds are used for training and one-fold is used for testing. Following are the classification algorithms tried out.

- Decision Tree
- Naïve Bayes
- K-Nearest Neighbor
- Support Vector Machine

Following are the reasons to use k-fold cross validation instead of using the traditional training/test data split [54].

- Utilizes all the data: Specially because the dataset used in this research is medium sized, using a 70:30 split will not utilize the data correctly and the size of the test dataset will be smaller. By performing cross-validation, we can utilize all our corpus data both for training and for testing while assessing our classification algorithm on instances it has not seen earlier. Specially with multiple labels or tag categories, having a mechanism like this was a must to make sure that tag values are distributed evenly across training/test data [54].

- Provides more metrics: When we build k distinct models using our algorithm and test it on k different test sets, we are able to be sure and certain in our algorithm's performance. If we depend on only one result, we cannot be sure of the result even if it performs well or even if it performs bad. It can perform well due to unbalanced nature of labels in the dataset or another bias. Therefore, to be certain about the performance of the model we build it is always good to go for cross validation. For instance, if the model accuracy is consistent across all the folds, then our dataset is performing quite well. With an approach like this we can compare metrics across folds and draw conclusions [54].

- Parameter finetuning: Most of the classification algorithms we tried required parameter tuning. We do it by attempting various values and picking the finest ones. So rather than having a different validation set, we can use one of our data splits for this purpose [54].

4.1.2 Confusion matrix

Confusion matrix is one of the most commonly used ways of evaluating classification model performance. When a classification-based approach is used for sequence tagging, each tag will be treated as a 2-class classification problem. Calculating the confusion matrix for the models trained will be really helpful in computing further informative matrices [55].

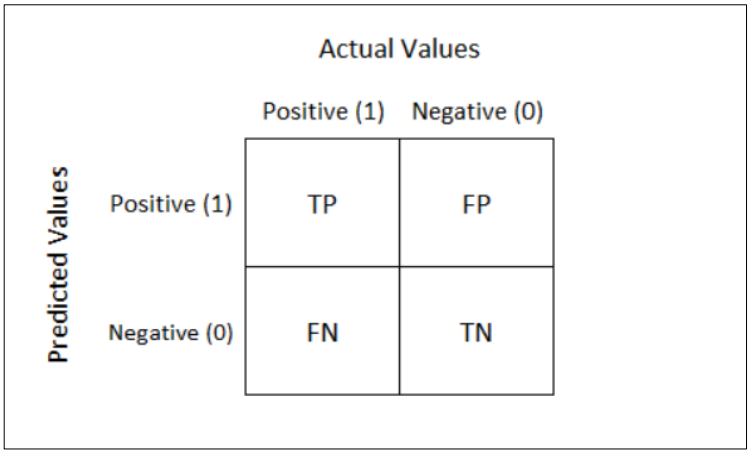


Figure 4.1: Confusion Matrix

Based on the above figures, three important metrics were calculated with the intention of evaluating model performance. They are,

- Precision
- Recall
- F1 Measure

Precision is how many are really positive out of all the positive classes predicted correctly. This figure should be high as possible [55].

$$Precision = \frac{TP}{TP + FP}$$

Recall is how much we predicted accurately out of all the positive classes. It too should be high as possible [55].

$$Recall = \frac{TP}{TP + FN}$$

If the precision is high and recall is low or the vice versa, it might be difficult to compare these figures and get an understanding of the actual accuracy. The idea of having F1 measure is drawing a comparative conclusion. It uses Harmonic Mean instead of Arithmetic Mean by penalizing the excessive values further [55].

$$F - Measure = \frac{2 * Recall * Precision}{Recall + Precision}$$

All these three measures were obtained and compared for each classifier algorithm to get an idea of the model accuracy in a comparative way.

4.1.3 Experimental setup

As mentioned earlier, four classification algorithms were trained with k-fold cross validation having 5 as the number of folds. Then for each fold confusion matrix was computed and precision, recall and f1 score values were obtained. Therefore, for each algorithm five precision, recall and f1 measures were computed. Then an average value was obtained to indicate the precision, recall and f1 score for each algorithm.

These measures were obtained for each tag in the dataset, to get an understanding on how the models are performing in terms of each tag. The main reason for looking into how the model is performing in each privacy attribute (Direct Identifiers/Quasi Identifiers/Sensitive Attributes) was the fact that model accuracy being highly impacted by the relatively high count of the tag 'NONE'. As any sentence contains a relatively high number of 'NONE' tags, there is a good chance that those words being predicted correctly, thus giving a pretty high overall accuracy. But this is not the actual result and it is caused by the bias mentioned above. Therefore, it was mandatory to get some measurements on individual attributes.

Then as an overall measure of the models, both macro average and weighted average of the accuracy was measured. Macro average portraits the accuracy without considering the distribution of labels in the dataset. That means macro average is independent of the classes or tags in the dataset. But weighted average is different from this and affected by the distribution of classes in the dataset.

Finally, all these measures were compared to get an idea about how each model is performing and based on that comparison, the best model was picked.

4.1.4. Experimental results

Table 4.2 summarizes the average values across 5 folds for precision, recall and f1 score values for each algorithm used. Looking at those figures it can be understood, how each classification model is performing in terms of each privacy attribute.

Algorithm	Macro Average	Weighted Average
Decision Tree	0.6122	0.91
Naïve Bayes	0.5664	0.91
KNN	0.6330	0.93
SVM	0.7245	0.94

Table 4.1: Average Accuracy Values for Each Algorithm

Decision Tree	Precision	Recall	F1-Score
DI	0.83	0.76	0.79
QIAGE	0.45	0.66	0.54
QIGENDER	0.40	0.68	0.50
QIREGION	0.37	0.41	0.39
QILANG	0.56	0.64	0.60
QIJOB	0.56	0.42	0.48
QIMARITAL	0.42	0.46	0.44
SA	0.74	0.63	0.69
NONE	0.97	0.97	0.97
Naïve Bayes	Precision	Recall	F1-Score
DI	0.65	0.77	0.70
QIAGE	0.30	0.40	0.34
QIGENDER	0.46	0.79	0.58
QIREGION	0.7	0.23	0.35
QILANG	0.6	0.58	0.59
QIJOB	0.11	0.23	0.15
QIMARITAL	0.54	0.8	0.64
SA	0.78	0.59	0.67
NONE	0.97	0.97	0.97
KNN	Precision	Recall	F1-Score
DI	0.76	0.78	0.77
QIAGE	0.45	0.55	0.50
QIGENDER	0.54	0.66	0.59
QIREGION	0.43	0.21	0.28
QILANG	0.67	0.56	0.61
QIJOB	0.55	0.4	0.46
QIMARITAL	0.84	0.79	0.81

SA	0.80	0.62	0.70
NONE	0.97	0.97	0.97
SVM	Precision	Recall	F1-Score
DI	0.85	0.75	0.80
QIAGE	0.50	0.58	0.54
QIGENDER	0.59	0.82	0.69
QIREGION	0.64	0.49	0.56
QILANG	0.75	0.67	0.71
QIJOB	0.85	0.50	0.63
QIMARITAL	0.96	0.74	0.84
SA	0.88	0.72	0.79
NONE	0.98	0.98	0.98

Table 4.2: Accuracy Details per Class

By analyzing the results, it is evident that Support Vector Machines has outperformed the other algorithms slightly. This is quite visible if we look at the macro average column of each attribute. Weighted average cannot be used for such a comparison, because that is affected by the high number of 'NONE' tag count.

Even if each identifier value in Table 4.1 is compared across algorithms, it is visible that SVM has outperformed other algorithms. KNN was performing as the second best, then Decision Tree algorithm at the third place and Naïve Bayes comes at the last place.

4.2 Mocking real world PPDP workflows

A set of experiments were designed to simulate how each component of the PPDP workflow can be integrated together and to understand the strengths and weaknesses of each module. Those experiments and their outcomes will be discussed in this chapter. A web application was developed so that any user can try out these experiments by him/herself.

4.2.1 Single tweet experiment

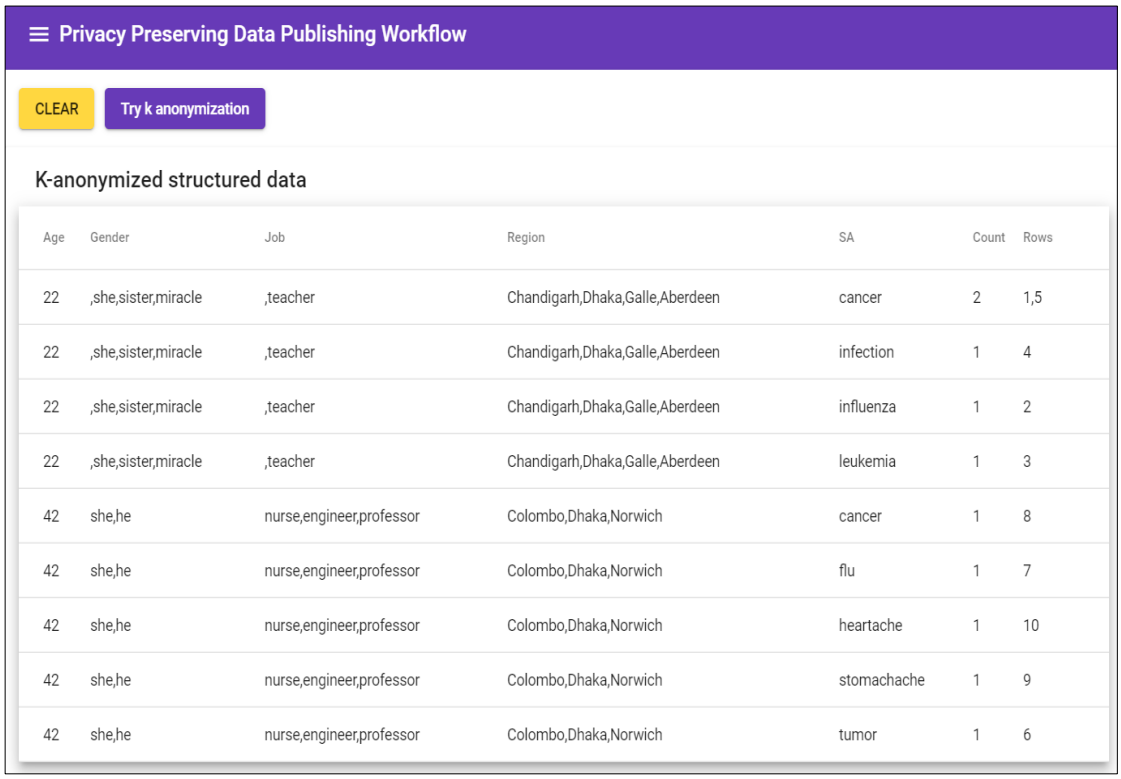
This allows the user to insert one tweet/sentence and perform tagging and anonymization on that specific tweet. The purpose of enabling that was to allow the user to try and experience the capabilities of the framework and how it operates. The process that is happening to a single tweet will be repeated for a list of tweets in a different environment. So, understanding what happens with one tweet and playing around it will be the entry point for the framework.

4.2.1 Multiple tweets experiment

The main intention of this experiment was to showcase the performance of the privacy attribute extractor and how it adopts to new samples. Therefore, a set of sentences were carefully built including a diverse range of direct identifiers, quasi identifiers, and sensitive attributes. First privacy attribute extraction was performed on this dataset and then both simple anonymization and k-anonymization was performed on the dataset. Privacy attribute extractor was performing with more than 90% accuracy compared to human annotation with this experimental dataset and it was performing well in predicting the tags of most of these new and unseen privacy attributes.

Then the annotated dataset was input into the anonymizer. Both simple anonymization and k-anonymization was performed on this annotated experimental data. It gave a good understanding on how data looks like when they were anonymized. As the expected result was known in advance, it was easy to figure out how the privacy attribute extractor and the anonymizer was performing. On the anonymized dataset, anonymization statistics and utility metrics (loss metric and discernibility metric) were calculated too.

Figure 4.2: Results from the experimental data set annotation



Age	Gender	Job	Region	SA	Count	Rows
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	cancer	2	1,5
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	infection	1	4
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	influenza	1	2
22	,she,sister,miracle	,teacher	Chandigarh,Dhaka,Galle,Aberdeen	leukemia	1	3
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	cancer	1	8
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	flu	1	7
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	heartache	1	10
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	stomachache	1	9
42	she,he	nurse,engineer,professor	Colombo,Dhaka,Norwich	tumor	1	6

Figure 4.3: Results from the experimental data set k-anonymization

4.2.3 Twitter live search experiment

This experiment is about creating a real time dataset from Twitter on the go. Feasibility to perform a keyword search on Twitter is provided based on a desired keyword. Then a dataset of 100 tweets will be extracted. This dataset can be used to perform the privacy attribute extraction, anonymization, and metric calculation on a completely unseen dataset.

By looking at the extracted privacy attributes, it is evident that the automatic tagger is capable of extracting a wide range of privacy attributes. In addition to that, anonymization (both simple anonymization and k-anonymization) of this dataset and calculating utility metrics on that can give us an understanding of how a real world PPDP operation can be.

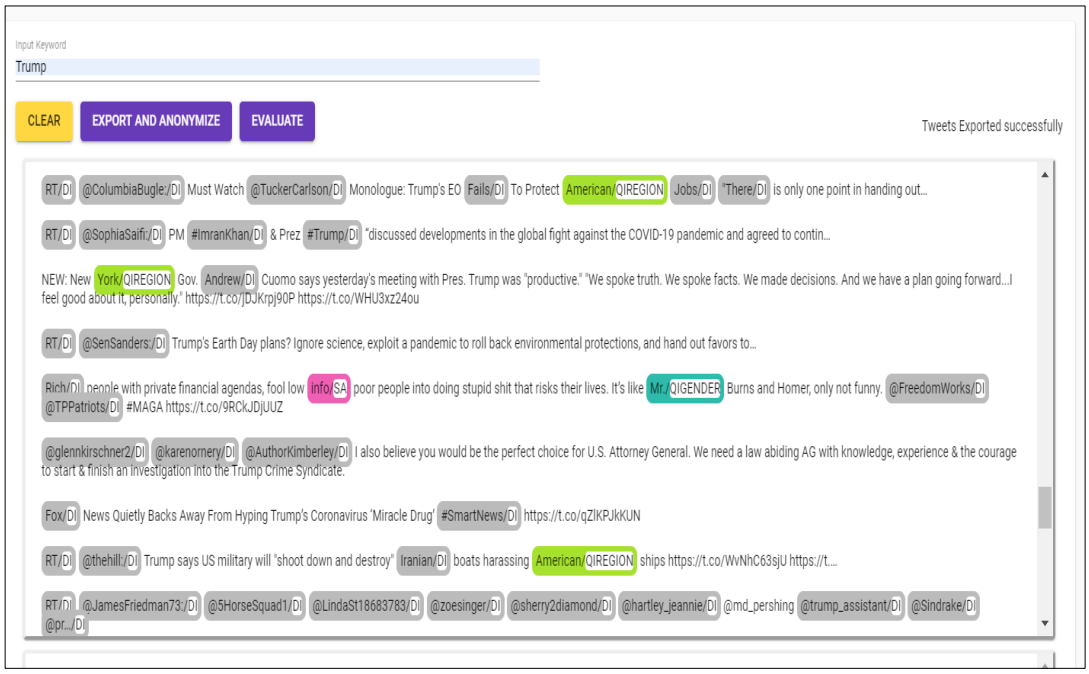


Figure 4.4: Results from the live twitter data set annotation

4.2.4 Usability evaluation of anonymized dataset

The next experiment is to verify whether the usability of the dataset is still protected even after it is subjected to a PPDP workflow. For this, a hate speech analysis problem was simulated using a public tweet set [56]. First the dataset was used to train a logistic

regression classifier to solve the above problem. Then the f1 score was computed for the trained model.

Then the same dataset was anonymized using simple anonymization and k-anonymization and then again, the logistic regression classifier was trained to perform sentiment analysis. The confusion matrix was computed again. The results are listed in table 4.3.

	F1-Score
Before Anonymization	0.5320
After Simple Anonymization	0.5284
After k-anonymization	0.5299

Table 4.3: Utility comparison after anonymization

4.3 Summary

This chapter summarizes the experiments designed to assess the performance of the machine learning models as well as to simulate the real word PPDP workflows. Results of the experiments prove that the overall framework is in a ready to be adopted state by a wider community.

CHAPTER 5: CONCLUSION

This chapter will bring up some concluding notes including a summary of the work done, research outcomes, research limitations and future work.

5.1 Summary

In summary, this research tries to implement a framework which enables privacy preserving data publishing with unstructured textual data. Specifically, the research is targeting social media data, due to the nature of social media data and the analytical value social media data has. It is highly likely for social media data to be used in disinformation campaigns targeting individuals. Twitter was used as the social media framework of interest in developing the framework – both to build models and evaluate the performance. Even though the framework is said to address unstructured data, it is implemented in a way where it can be extended to structured data as well. And also, any other platform can use the framework, even though it was implemented having Twitter as a use case.

The core framework contains components to extract privacy attributes from textual data, anonymize extracted attributes and calculate anonymization metrics on anonymized dataset including utility metrics. Privacy attribute extraction happens through a classification-based sequence tagger. Five different algorithms were tried to pick the best sequence tagging approach and out of them Support Vector Machine had the best performance. It had an average F1 score value of 0.72 across all the tag types. In order to train the automatic annotator, a data corpus including tweets with manually annotated privacy attributes were created and this corpus might be the first such corpus in the research community. Other than that, well known privacy techniques and models from the research world and literature is integrated into the framework too. To enhance the capabilities of the framework, ability to compute statistics and metrics on any anonymization task performed is also provided. Utility metrics like loss metric and discernibility metrics are two such metrics.

When experimental design is considered, the main objective was to assess the performance of the automatic privacy attribute annotator. Experiments were designed

to achieve that task. And a separate set of experiments were designed to simulate real world PPDP workflows using different types of datasets. One other important experiment was to compare the usability of data, before and after being subjected to a PPDP workflow. The experiment proved that the usability is not affected largely after anonymization.

The proposed solution helps maintaining the data utility in different ways.

- Identifying/Anonymizing only the related quasi identifiers without going for a generic named entity recognition
- Identifying quasi identifiers which are highly used in the social media domain
- Using a variety of quasi identifiers help to come up with separate anonymizations schemes per attribute category
- Preserving hashtags
- Providing the ability to perform k-anonymity so that the user can decide on the required k value

Overall, this framework can be very helpful for any party who involved in collecting, publishing, and analyzing data specially if the data involves sensitive personal information. Social media is only one example for such a case, but the framework can be extended to many other domains and bodies.

5.2 Research outcomes

There were a set of valuable outcomes of this research. Especially due to the fact that privacy preserving data publishing has not gained a huge attention in the context of unstructured data, a framework like this and the underlying approach can be very valuable to the research community. Following are the main research outcomes.

- Coming up with an end to end framework, that includes all the necessary components in a privacy preserving data publishing workflow
- Coming up with a mechanism to support privacy preserving data publishing of unstructured and textual social media data

- Creating a corpus of tweets which includes tweets that are annotated with privacy related attributes. This can be the first or one of the very few publicly available such dataset.
- Coming up with a demonstration platform, which can showcase the capabilities of the framework, so that any party willing to use it can try it and play around before using
- Paper publication at LREC 2020, STOC Workshop (First International Workshop on Social Threats in Online Conversations)

5.3 Research limitations

The main objective was this research was to come up with an end to end framework, which can realize the idea of privacy preserving data publishing of unstructured, textual social media data. Even though the research outcomes were comparatively dominating, there were a couple of limitations in the research as well.

One of such limitations resides in the way the tweets are mapped to structured data when k-anonymization is performed. The data transformation to structured data assumes that one tweet is directly linked to one individual. Having a mechanism to separate out the links between individuals within a single tweet could have improved the k-anonymization process more than its current version.

The other limitation also lies within the k-anonymization. And that happens when a tweet contains more than one direct identifier, quasi identifiers or sensitive attributes of the same category. In such a scenario, only the last attribute value will be used in building the structured dataset. For example, if a tweet contains more than one QIREGION tags, the framework will use only the last region value when building the structured dataset. But both the attribute values will be anonymized.

5.4 Future work

This research can be extended in a couple of directions in order to enhance the adoption percentage and to support a wide range of identifiers.

1. Identify more quasi identifiers and embed them into the framework. Other than the quasi identifiers we used, there are different quasi identifiers that we can integrate, which will enhance the usage of the framework. Date of Birth, level of education and marital status are some of those additional quasi identifiers.
2. The data corpus can be further enriched with the newly recognized quasi identifiers.
3. More advanced anonymization techniques can be integrated into the system and it can be exposed in a configurable way for the user to select the anonymization mechanism. For instance, more advanced functionalities like user defined generalization hierarchies can be supported.
4. Support personalized privacy by allowing the user to select the sensitive attributes and other quasi identifiers because privacy is subjective from person to person.

5.5 Discussion

This research was conducted to come up with a privacy preserving data publishing framework that can be utilized by different parties to protect personal information captured in their data. The proposed system provides a flexible way of integrating different pieces of the PPDP workflows into a single pipeline using a loosely coupled plug and play architecture. This framework supports the capability to extract privacy attributes from unstructured data, anonymize them using simple or k anonymization and the calculate evaluation metrics for a single PPDP workflow.

A RESTFUL API based loosely coupled architecture was adopted and when implementing the framework with the purpose that any interested party can consume these API endpoints separately or in combination with each other in their systems. A sample demonstration framework is also implemented so any interested party can use that to play around and understand how these workflows can be implemented by connecting different pieces. API endpoints are well documented using Swagger, so anyone interested can go through it and understand the APIs. The Swagger documentation is included in the methodology chapter. This code base of the entire

framework and the dataset use is publicly available in this git repository to be used by anyone - https://gitlab.com/PrasadiApsara/ppdp

This framework was implemented in a way it can be easily utilized and extended as needed by any other party. As this is focusing on social media data specifically, social media platforms like Facebook, Twitter and Instagram can easily utilize this framework to protect their unstructured/textual data consuming the open REST API, mainly due to the fact it is platform independent. As the overall architecture is based on the concept of separation of concerns and each piece of the anonymization workflow are loosely coupled, they can even extend and modify the offered features too. For example, implementing l-diversity privacy model to the workflow is easy as it is just about coming up with an implementation and exposing as an endpoint.

This research was recognized and published under the Language Resource Evaluation Conference (LREC) – First International Workshop on Social Threats in Online Conversations (STOC) for its validity and the research paper is available in the Association for Computational Linguistics (ACL) Anthology - https://www.aclweb.org/anthology/2020.stoc-1.4/

REFERENCES

[1] A. Alaphilippe, A. Gizikis, C. Hanot, “Automated tackling of disinformation”, 2019, [Online]. Available: https://www.europarl.europa.eu/RegData/etudes

[2] General Data Protection Regulation [Online]. Available: https://en.wikipedia.org/wiki/General_Data_Protection_Regulation

[3] The “localisation” of Russian citizens’ personal data [Online]. Available: https://home.kpmg/be/en/home/insights/2018/09/the-localisation-of-russian-citizens-personal-data.html

[4] B. B. Mehta, U. P. Rao, “Privacy preserving unstructured big data analytics – issues and challenges”, in Proc. International Conference on Security and Privacy, Nagpur, India, 2015, pp. 120-124.

[5] R. Mendes, J. P. Vilela, “Privacy-preserving data mining: Methods, metrics, and applications,” in IEEE Access, 2017, pp. 10562–10582.

[6] P. Usha, R. Shriram, S. Sathishkumar, “Multiple sensitive attributes-based privacy preserving data mining using k-anonymity” in Int. J. Sci. Eng. Res. 5(4), 2014

[7] A. Kaur, “A hybrid approach of privacy preserving data mining using suppression and perturbation techniques,” in Proc. IEEE International Conference on Innovative Mechanisms for Industry Applications, ICIMIA 2017, India, 2017, pp. 306–311

[8] D. Kumari, Y. Vineela, T. Krishna, B. Kumar, “Analyzing and performing Privacy Preserving Data Mining on medical databases” in Indian Journal of Science and Technology. 2016 May; pp. 1–9.

[9] L. Sweeney, “Achieving k-Anonymity Privacy Protection Using Generalization and Suppression” in International Journal on Uncertainty, Fuzziness and Knowledge based Systems 10(5), 2002, pp. 571–588

[10] L. Zhang, W. Zhang, “Generalization-based privacy-preserving data collection”, in: Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery, DaWak, 2008.

[11] S. Hajian, J. Domingo-Ferrer, O. Farr`as, “Generalization-based privacy preservation and discrimination prevention in data publishing and mining” DMKD, 2014, pp. 1158–1188

[12] W. Yu, P. Lv, N. Chen, "Multi-Attribute Generalization Method in Privacy Preserving Data Publishing," in 2010 2nd International Conference on E-business and Information System Security, Wuhan, 2010, pp. 1-4.

[13] W.K. Wong, N. Mamoulis, D.W.L. Cheung, “Non-homogeneous generalization in privacy preserving data publishing” In SIGMOD, 2010, pp. 747–758

[14] A. S. M. Hasan, Q. Jiang, J. Luo, C. Li, L. Chen, “An effective value swapping method for privacy preserving data publishing” in Security and Communication Networks 9, 3219, 2016

[15] A. Richard, “Controlled Data-Swapping Techniques for Masking Public Use Micro Datasets”

[16] S.E. Fienberg, and J. McIntyre, “Data Swapping: Variations on a Theme by Dalenius and Reiss”, in Journal of Official Statistics, 9, pp. 383-406.

[17] V.S. Susan, T. Christopher, “Anatomisation with slicing: a new privacy preservation approach for multiple sensitive attributes” Springerplus 5(1), 964, 2016

[18] T. G. Marathe, M. N. Raverkar, S.G. Suryawanshi, M.F. More, “Preserving the Privacy of User by Using Anonymization Techniques” in International Journal of Sustainable Development Research, 2017

[19] R. Oksvort, “A Prototype for Learning Privacy-Preserving Data Publishing”, 2017

[20] B. Thuraisingham, M. Kantarcioglu, L. Liu, “Perturbation based privacy preserving data mining techniques for real-world data”, Doctoral Dissertation, 2008

[21] N. Patel, S. Patel, “A Study on Data Perturbation Techniques in Privacy Preserving Data Mining” in International Research Journal of Engineering and Technology, 2015

[22] B. C. Chen, D. Kifer, K. LeFevre, A. Machanavajjhala , “Privacy-preserving data publishing,” in Proc. Foundations and Trends in Databases Conference, 2009, pp. 1 – 167.

[23] P. Samarati, L. Sweeney, “Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and cell suppression”, Technical report, SRI International, 1998.

[24] G. Navarro-Arribas, V. Torra, A. Erola, J. Castellà-Roca, “User k-anonymity for privacy preserving data mining of query logs”, Information Processing and Management 48 (3), 2012, pp. 476–487.

[25] S. Ni, M. Xie, M, Q. Qian, “Clustering Based K-anonymity Algorithm for Privacy Preservation” in International Journal of Network Security, 2017, pp. 1062–1071.

[26] A. Machanavajjhala, J. Gehrke, D. Kifer,, M. Venkitasubramaniam., “l-diversity: Privacy beyond k-anonymity”, in Proc. 22nd International Conference on Data Engineering (ICDE). IEEE Computer Society, 2006

[27] N. li, T. Li, S. Venkatasubramanian, “t-Closeness: Privacy Beyond k-Anonymity and l-Diversity”, in IEEE 23rd International Conference on Data Engineering, 2007

[28] Z. Huang, “Privacy-Preserving Algorithms for Genomic Data”

[29] C. Uhlerop, A. Slavković, S. E. Fienberg, “Privacy-Preserving Data Sharing for Genome-Wide Association Studies”, 2015

[30] J. Gardner and L. Xiong, “An integrated framework for de-identifying heterogeneous data”, in Proc. Data and Knowledge Engineering, 2009, pp. 1441-1451

[31] K. Liu, K. Das, T. Grandison, and H. Kargupta. “Privacy-preserving data analysis on graphs and social networks”, In H. Kargupta, J. Han, P. Yu, R. Motwani, and V. Kumar, editors, Next Generation Data Mining. CRC Press, 2008

[32] J. Vadisala and V. K. Vatsavayi, “Challenges in Social Network Data Privacy” in International Journal of Computational Intelligence Research (IJCIR), vol -13, 2017, pp. 965-979

[33] C. Liu, P. Mittal. “Linkmirage: Enabling privacy-preserving analytics on social relationships”, in NDSS, 2016, pp. 21-24

[34] A. Meyerson and R. Williams, “On the complexity of optimal k-anonymity,” in Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS), 2004.

[35] V. S. Iyengar, “Transforming data to satisfy privacy constraints,” in ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002.

[36] R. Bayardo and R. Agrawal, “Data privacy through optimal k-anonymity,” in Proceedings of the 21st International Conference on Data Engineering (ICDE), 2005.

[37] K. LeFevre, D. DeWitt, and R. Ramakrishnan, "Workload-Aware Anonymization," in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2006.

[38] X. Xiao and Y. Tao. "Personalized privacy preservation" In Proceedings of ACM Conference on Management of Data (SIGMOD'06), 2006, pp. 229–240

[39] D. P. M. Kumar and Y. P. Gowramma. "Development of Sensitivity Classification Approach for Personalized Privacy Preservation in Data Publishing (PPPDP)" in International Journal of Innovative Research in Computer and Communication Engineering, 2017

[40] S. Chakraborty, C. Shen, K. R. Raghavan, Y. Shoukry, M. Miller, M. B. Srivastava, "ipShield: a framework for enforcing context-aware privacy" in Proceedings of USENIX Symposium on Networked Systems: Design and Implementation (NSDI) ; 2014; Seattle, WA, pp. 143–156 .

[41] Big Data and the Challenge of Unstructured Data [Online]. Available: https://www.ciklum.com/blog/big-data-and-the-challenge-of-unstructured-data/

[42] V. Vincze, R. Farkas, "De-identification in Natural Language Processing", in proceedings of 37th Convention of Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2014, pp. 1300-1303.

[43] L. Sweeny, "Replacing Personally Identifiable Information in Medical Records, the Scrub System", in the Journal of the American Medical Informatics Association, 1996.

[44] I. Neamatullah, M. Douglass, L. Lehman, A. Reisner, M. Villarroel, W. Long, G. Clifford, "Automated de-identification of free-text medical records" in BMC Medical Informatics and Decision Making, 2014.

[45] R. Motwani, S. Nabar, "Anonymizing unstructured data" [ONLINE]. Available: https://arxiv.org/pdf/0810.5582.pdf

[46] UK Data Service, [ONLINE]. Available: https://bitbucket.org/ukda/ukds.tools.textanonhelper/wiki/Home

[47] H. Vico, D. Calegari, “Software Architecture for Document Anonymization” in Electronic Notes in Theoretical Computer Science, 2015, pp.83-100.

[48] B. Kleinberg, M. Mozes, Web-based Text Anonymization with Node.js: Introducing NETANOS (Named entity-based Text Anonymization for Open Science) in the Journal of Open Source Software

[49] X. Carreras, L. Marquez, L. Padro´, “A Simple Named Entity Extractor using AdaBoost”, in Proceedings of Proceedings of CoNLL, 2003

[50] Industrial-Strength Natural Language Processing [ONLINE]. Available: https://spacy.io/

[51] Overview of Classification Methods in Python with Scikit-Learn [ONLINE]. https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/

[53] K. LeFevre, D. DeWitt, R. Ramakrishnan, “Mondrian Multidimensional K-Anonymity”, in Proceedings of 22nd International Conference on Data Engineering (ICDE), 2006

[54] 5 Reasons why you should use Cross-Validation in your Data Science Projects [ONLINE]. Available: https://towardsdatascience.com/5-reasons-why-you-should-use-cross-validation-in-your-data-science-project-8163311a1e79

[55] Understanding Confusion Matrix [ONLINE]. Available: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

[56] Practice Problem: Twitter Sentiment Analysis [ONLINE]. Available: https://datahack.analyticsvidhya.com/contest/practice-problem-twitter-sentiment-analysis/#ProblemStatement

[57] Twitter Sentiment Analysis [Online]. Available: https://www.kaggle.com/paoloripamonti/twittersentiment-analysis