FEDERATED LEARNING TO BUILD SENTIMENT ANALYSIS MODELS FOR AMAZON REVIEW DATASET WITHOUT LABELS

by

Ritu Kadve, B.Tech

THESIS

Presented to the Faculty of

The University of Houston-Clear Lake

In Partial Fulfillment

Of the Requirements

For the Degree

MASTER OF SCIENCE

in Computer Science

THE UNIVERSITY OF HOUSTON-CLEAR LAKE

MAY, 2023

FEDERATED LEARNING TO BUILD SENTIMENT ANALYSIS MODELS FOR AMAZON REVIEW DATASET WITHOUT LABELS

by

Ritu Kadve, B.Tech

APPROVED BY

_____

Kewei Sha, Ph.D., Chair

_____

Yalong Wu, D.Sc., Committee Member

_____

Wei Wei, Ph.D., Committee Member

APPROVED/RECEIVED BY THE COLLEGE OF SCIENCE AND ENGINEERING:

_____

David Garrison, Ph.D., Associate Dean

_____

Miguel Gonzalez, Ph.D., Dean

Dedication

I dedicate this thesis to my family back in Hyderabad, India – my parents Rajeev Kadve and Trusha Kadve, my little sister Resha Kadve, my little brother Rio Kadve, and my granny Suvarna Kadve, and my extended family from Maharashtra, India – My grandparents Hemanth Patel and Jyothi Patel, my Mama Mami Akshay Patel and Rucha Patel and my little brothers Asit Patel and Olive Patel. I would also like to thank my uncle from Canada – Som Vembar for supporting me.

Thank you to you all for constantly encouraging and believing in me during my Master's studies. It would not have been possible without your blessings.

Acknowledgements

First and foremost, I would like to thank Dr. Kewei Sha, my thesis advisor who saw the potential in me right from the first semester. His valuable feedback, suggestions, encouragement, and patience have helped me succefully complete my thesis.

I would also like to express my gratitude to Dr. Yalong Wu, my thesis committee member for his constant support and guidance throughout the thesis. I am grateful to Dr. Wei Wei, my thesis committee member for her constructive input.

Special thanks to Parth Trada for encouraging and motivating me to do this study and supporting me in every step of it. It would not have been possible without his help. I cannot thank him enough.

Last but not the least, I would like to thank my best friends – Shruthi Natte and Shravya Bhatt who are also my roommates for constantly cheerleading me and boosting my morals during my Master's studies.

I sincerely express my gratitude to each one of you and I am deeply grateful for all the inspiration.

ABSTRACT

FEDERATED LEARNING TO BUILD SENTIMENT ANALYSIS MODELS FOR AMAZON REVIEW WITHOUT LABELED DATASET

Ritu Kadve
University of Houston-Clear Lake, 2023

Thesis Chair: Dr. Kewei Sha

In Natural Language Processing, one of the most popular tasks is sentiment analysis which aims to predict the sentiment of a text. It has many practical applications in industries such as marketing and customer service. Performance of sentiment analysis models play a significant role to the success of these applications. To achieve a high accuracy, sentiment analysis usually trains analytical models based on labeled datasets, preferably to large-scale labeled dataset. However, large-size labeled dataset may not be available because of the high-cost in labeling. Therefore, researchers study alternative approaches aiming to learn high accurate and reliable models based on small-scale labeled datasets or using other existing labeled datasets from different categories. A centralized model is a machine learning model that utilizes a large dataset stored on a central server to perform sentiment analysis. Training a centralized model on a small, labeled dataset can result in inaccurate or incomplete predictions. While processing

labeled datasets of different categories on a centralized platform, it also comes with many challenges such as data heterogeneity, bias towards the categories that are overrepresented, requirement of large amount of computational power and resources, and the availability of good amount of labeled data for training. In addition, it is difficult to select appropriate data categories to train a reliable model for the new category. In this thesis, we propose a federated learning approach to overcome these challenges.

Federated Learning (FL) is a type of decentralized Machine Learning (ML) that lets us train data analytical models on local data without transferring data to a central server. When Federated Learning is applied to sentiment analysis, one server and multiple clients collaborate to train a reliable and accurate sentiment analysis model. In our scenario, each client trains a local sentiment analysis model based on a labeled review dataset of a specific category, and the server makes use of the FedAvg algorithm to aggregate the parameters from the trained client models to build a global model for the new category that has no available labeled dataset. We evaluate the performance of our design based on a prototype implementation using Amazon review datasets. Compared with the centralized sentiment analysis, the proposed FL-based sentiment analysis performance is 10% better. This validates the potential of federated learning in training better data analytical models for categories with no large-scale labeled datasets.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER I:

INTRODUCTION

1.1 Background and Significance

Sentiment analysis is a specialized technique of Natural Language Processing (NLP) that helps us identify and classify the information that has opinions or emotions expressed in the text. The main objective of sentiment analysis is automatically classifying the opinions expressed in a sentence as either positive, negative, or neutral. This task is also known as sentiment mining, opinion mining, or emotion Artificial Intelligence (AI) [1]. Sentiment analysis is used in various applications like product review analysis [2], customer feedback analysis [3], or social media monitoring [1].

The most common use of sentiment analysis is to analyze the reviews. Amazon review dataset is considered to be one of the most popular and rich sources of user-generated dataset. It contains millions of reviews for various product categories ranging from books, software, electronics, digital music, prime pantry to fashion. Analyzing the reviews can be considered an important aspect for a business as it provides valuable insights and customer's point of view with respect to the business's products and services. While a successful business requires sentiment analysis to be accurate, reliable, and scalable which usually demands a high-quality labeled dataset. However, labeling can be time-consuming and an expensive task, especially when a large-scale labeled dataset is preferred. When there is lack of labeled training dataset, it is difficult to build an accurate sentiment analysis model. Such models require a significant amount of training data to accurately capture the nuances of language and context.

Sentiment analysis is subjective and context dependent. This means that when we train a model on a small set of labeled data, we may not be able to capture the full range of the sentiments that are expressed in the review category without labeled dataset. This

can lead to a biased model or incomplete results as the model may not accurately reflect the sentiments of reviews [24]. An alternative approach is to train sentiment analysis models for a review category that has no available labeled dataset by using existing labeled datasets of other categories. Centralized sentiment analysis or traditional method of performing sentiment analysis on different categories of review dataset possess challenges as well. As Amazon has various categories and each category has its own language and product characteristics, it is difficult to train a central model that fits all categories. The model's performance depends on the contexts of the reviews, the kind of words used in the review and domain of the reviews. Traditional approaches based on labeled data of other categories might not have a high accuracy. To address these problems, we propose Federated Learning, which is a decentralized approach where multiple parties collaborate to train a global model without exchanging any data.

Federated learning (FL) was developed in 2016 [4]. It is an approach where the training data does not leave the client device and is used to train models locally. The main idea behind federated learning is that the data remains private to each client, and the server only receives the local model's parameters. This approach is extremely useful in sectors that require data privacy like health care, banking, and financial industries. New applications of federated learning are yet to be explored.

We aim to improve the performance of sentiment analysis models for datasets that have no available labeled training set by training local models and then global model using other categories of data. In this way, a global model is trained over a more diverse and representative data, and most importantly on different product categories. We hope the trained model can be more accurate and generalizable. It can also be more efficient, because the data does not have to be transmitted over the network, which can reduce the amount of time and resources required for training. As the model is trained on multiple

datasets, it can be more reliable too [4]. Finally, Federated Learning helps to safeguard the confidentiality and security of the client data.

In this thesis, we analyze the effectiveness of our approach by comparing it with the model trained using the traditional centralized sentiment analysis method and verify that the federated learning approach provides better performance when it comes to various review domains.

1.2 Research Goals and Questions

This thesis has the following research goals. Firstly, we want to train a sentiment analysis with good accuracy for datasets that have no available labeled training set. Secondly, we want to explore Federated Learning to achieve the above goal, i.e., building a global model based on local models trained from labeled dataset of different categories. We want to evaluate the effectiveness and efficiency of the Federated Learning approach in comparison to the traditional sentiment analysis approach. Additionally, we evaluate the performance of the proposed model with different number of categories of review datasets. By exploring the impact of the number of datasets and the combinations of categories used for training, we can identify the optimal setup for improving the performance of the global model. Moreover, we will examine the performance of the Federated Learning-based solution by training the model with a small dataset. We try to understand if the size of the datasets affect the model's performance. Finally, we aim to address the problem of data heterogeneity by distributing the review data across the clients. In summary, we aim to answer the following research questions:

1. Does the global model trained using Federated Learning perform better than a model trained using traditional sentiment analysis approach?
2. How will the model trained on small dataset perform on a larger dataset of same category?

3. Does the number of datasets and combinations of categories used for global model training affect the performance of global model?
4. Will creating random datasets improve the accuracy of our approach?
5. How does the model perform for sentence-based dataset and paragraph-based dataset?

1.3 Contributions of the Research

This research provides a significant contribution to the field of sentiment analysis by adopting a Federated Learning approach. Firstly, Federated Learning is rather unexplored in the field of natural language processing [4], particularly in sentiment analysis on review dataset, while it is more common for applications like image classification, speech recognition, etc. Secondly, in this study, we compare the performance of federated sentiment analysis to that of traditional centralized sentiment analysis. We can identify the benefits of applying the decentralized sentiment analysis over the centralized sentiment analysis, especially in terms of the unavailability of training dataset and generalization across different categories of data. Lastly, we propose a mechanism for training a global model of datasets without labeling it, which reduces the cost and time required for preparing labeled training dataset needed by supervised learning.

1.4 Research Design and Result

1.4.1 Research Design

In this study, we focus on investigating the potential of Federated Learning for sentiment analysis on unlabeled Amazon review data. Centralized platform to train analytical models from various categories of review datasets can have following problems:

1. *Data Heterogeneity* – There are several categories of product reviews on Amazon. Each category of review has its own review format, specific words, and product characteristics. This makes it difficult to have a single model that fits all categories.
2. *Bias* – The model can be biased if it is trained on a centralized dataset. This is because of overrepresented or underrepresented review categories.
3. *Scalability* – Processing a large, centralized dataset for training a model requires large computational power and resources.
4. *Labeled Dataset* – To train a single model that fits various categories of datasets, the model should be trained on a big size of labeled dataset. Labeled datasets are not easily available.

We aim to avoid above problems by designing a Federated Learning-based approach. Figure 1.4 presents the main idea of Federated Learning-based sentiment analysis for Amazon review. There two main components in Federated Learning– client and server. In Figure 1.4, each client has a category of review which is represented as Review data 1, Review data 2, and so on, and each local model is trained based on a particular dataset. The local model's parameters are sent to the server. The server aggregates these parameters using FedAvg algorithm and employs them to update the global model. This process is repeated iteratively until the global model converges. This architecture enables the training of machine learning models on distributed data sources without the need for centralized data storage. This thesis contributes a novel approach in the field of sentiment analysis on unlabeled Amazon review data and evaluates its effectiveness.

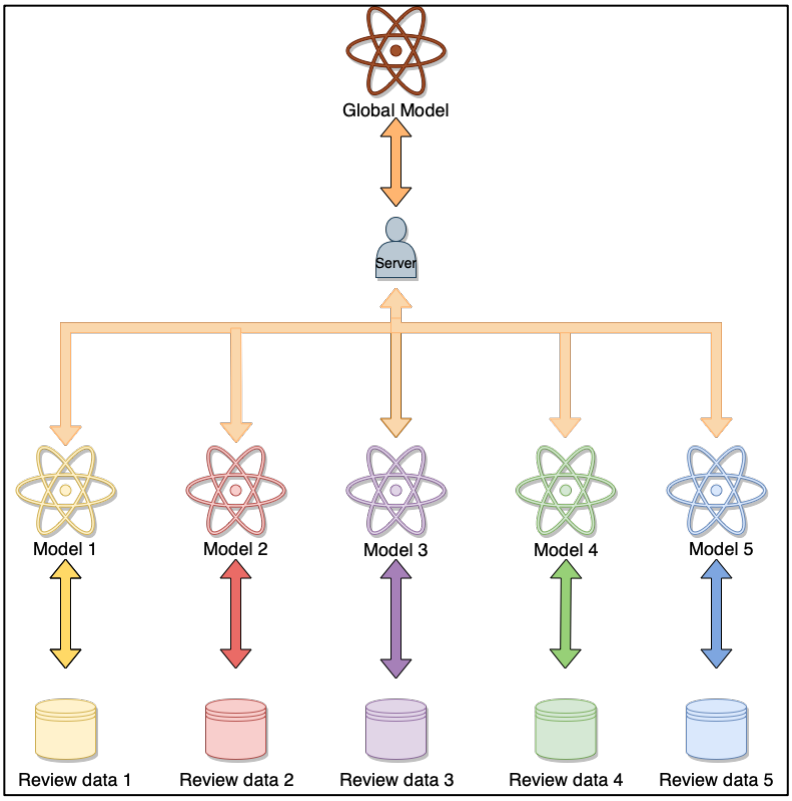

Figure 1.4 Architecture of the proposed approach

1.4.2 Result

We find that when training a sentiment analysis model for a dataset without labeled training set, the proposed Federated Learning-based approach generates a global model with an Area Under Precision-Recall Curve (AUPRC) score of 0.887, which is 10% better than the model generated by the traditional central approach with an AUPRC score of 0.800.

We find that using combinations of review categories to train the global model performs fairly well across different combinations. The highest AUPRC score of 0.89 is achieved for the combination of Software and Digital Music datasets, stating that certain combinations of review dataset used for training can result in a good sentiment analysis model. We also answer the research questions in Chapter 1.2 by well-designed experiments.

1.5 Organization of Thesis

In this chapter we provide the background and significance of our thesis, followed by the motivation of this study. Chapter II presents the literature review about Federated Learning in text mining and various federated learning algorithms that are relevant to sentiment analysis. Chapter III describes the datasets used in our study and the process to clean and get the dataset ready for analysis. In chapter IV, we illustrate the system architecture and the steps to train federated sentiment analysis models, which includes a detailed explanation of the local model and global model. We end this chapter by covering the practical implementation of Federated Learning (FL), which involves the creation of a federated dataset and the model training process using the federated approach. Chapter V discusses the performed experiments and the evaluation results. Chapter VI indicates the future work and Chapter VII concludes the thesis.

CHAPTER II:

LITERATURE REVIEW

2.1 Federated Learning in NLP

As Federated Learning was recently developed, its potential in the field of Natural Language Processing (NLP) remains largely unexplored. One of the real-world implementations of Federated Learning is the "Hey Siri" wake up word detection in iPhones [2, 4]. Guliani et al. discusses the application of Federated Learning to train speech recognition models. They propose an approach that optimizes the cost of training using Federated Learning [18]. Smith et al. explores multi-tasking applications such as image processing, speech-recognition using Federated Learning. They developed Mocha, a novel systems-aware optimization framework for federated multi-task learning and highlight the importance of using Federated Learning in multi-tasking while preserving the user's privacy on personal devices [22]. Federated Learning is a type of distributed Machine Learning. A simple way to understand the difference is that Federated Learning is "decentralized training over decentralized data" [10]. Hard et al. worked on increasing the predictive capabilities of mobile keyboards using Federated Learning approach. Here, Federated Learning was used to avoid the requirement of sharing user's personal mobile data on a centralized platform. They analyze this approach on a large-scale data to show that the accuracy can be maintained while preserving the privacy of the user [20]. T. Li et al. highlight the importance of Federated Learning in various domains such as healthcare, finance, and Internet of Things. They go over the opportunities of Federated Learning in these domains and advantages of using Federated Learning approach [13].

H. B. McMahan et al. focus on developing a communication – effective approach to train deep learning models on decentralized data. The authors compare Federated Learning approach with the traditional centralized methods to show that Federated

Learning has promising results for training deep learning models when we have decentralized data. They also discuss about communication overhead and privacy preservation [16]. To communicate the parameters from local models to the global model, Xinghua Zhu et al. used FedAvg (created by McMahan et al. [8]) for-text categorization [7]. Federated Learning has become popular because of various reasons, mainly because it solves data privacy issues. Advances in Deep Learning approaches can be used by domains that require privacy preservation with the help of Federated Learning [8]. Hilmkil et al. modified the parameters using a newer version of FedAvg [8]. Because large models cannot be used in tiny local devices, Sattler et al. used knowledge distillation to convey information when the local model size was small [10].

2.2 Federated Learning for Sentiment Analysis

Liang W et al. used Federated Learning Edge Network (FLEN) to tackle the web data related to global pandemic Covid19 data for sentiment analysis. They use FLEN to overcome the data privacy challenge. Their approach outperforms traditional approach in terms of accuracy, data privacy and efficiency. FLEN can be useful for domains similar to healthcare, such as finance where data-privacy is important [27]. Federated Learning via Model Distillation (FedMD) was employed by Tsankova P. et al. for the purpose of detecting sentiment in tweets while maintaining data privacy. FedMD needs two datasets—a global dataset and a private dataset—from separate sources. They focus on model personalization and data-heterogeneity by introducing client-specific models and collaborative learning [11]. In our thesis, we implement the FedAvg algorithm on review categories to build a global model to predict the sentiments of category without labeled dataset.

CHAPTER III:

DATA COLLECTION

3.1 Amazon Review Data

Due to its scale and diversity, the Amazon review dataset has emerged as one of the most popular datasets for sentiment analysis and natural language processing applications. The current Amazon review dataset contains 130 million reviews from 1995 to 2015 and a lexicon of nearly 200,000 new words. The star ratings for the reviews range from 1 to 5, with 5 representing the highest rating.

The dataset is frequently used for developing and testing sentiment analysis models, which identify whether a review is positive or negative based on its sentiment. The dataset is also utilized for various other NLP applications, including topic modeling, text classification, etc. It is important to note that the Amazon review dataset does not include any personal identity information about the reviewers, or the products being evaluated owing to privacy concerns. The dataset has also undergone preprocessing to get rid of any potentially harmful material, such as profanity or hate speech.

Moreover, we need datasets that are different from each other in terms of content and dialects. In order to improve the performance of Machine Learning models, it is important to have access to datasets that exhibit significant variability in terms of language and content used within them. Such diversity in datasets helps the Machine Learning models to learn and generalize sentiments accurately. Amazon review dataset has variety of products review that are different in terms of review contexts and the size of the dataset. This makes Amazon review dataset a great fit to our study.

3.2 Description of Dataset

3.2.1 Dataset from SNAP

We are using the Amazon review dataset available on the SNAP website [35]. Choosing from various topics, we select four different categories, including Health & Beauty Dataset, Digital Music Dataset, Musical Instrument Dataset, Industry & Science Dataset and Software Dataset. The review categories Digital music and Musical instruments might have few similarities with respect to the word vocabulary used to describe music and certain language slangs. The selection of these topics was based on how different they are from each other and to also have categories that have few similarities. This helps us understand how Federated Learning approach works in case of extremely different datasets as well as the datasets that have few similarities among them. Table 3.1 gives an overview of the different categories of reviews.

The selected categories are used to study the benefits of using Federated Learning to train a global model using available labeled training data. This trained global model will then be used to predict the sentiments of category which lacks training data. These categories of reviews are paragraph-based reviews. Semeval et al. analyzed why using longer texts (4-8 sentences) led to better accuracy for sentiment classification compared to using shorter texts (1-3 sentences). They suggest that this is because longer texts provide more context for understanding the sentiment of the text, which is particularly important in the case of short social media messages like tweets [29].

Table 3.1 Overview of Categories

Category	Positive Reviews	Negative Reviews	Total Reviews
Beauty	14,981	5019	20,000
Digital Music	12,000	8,000	20,000
Musical Instruments	10,470	9,530	20,000
Software	7,924	12,076	20,000
Industry & Science	11,957	8,043	20,000

From Table 3.1, the categories show imbalance in number of positives and negatives reviews. For example, the Software category has significantly more negative reviews compared to Beauty category.

The review in the dataset includes information such as the reviewer's ID, the product ID, the rating (on a scale of 1-5 stars), and the text of the review itself. In addition, the dataset includes metadata such as the date of the review and whether the review was marked as "helpful" by other users. The dataset is of the format JSON. A sample from the JSON review is given in Figure 3.1.


```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.
He is having a wonderful time playing these old hymns.  The music  is
at times hard to read because we think the book was published for
singing from more than playing from.  Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```


Figure 3.1 A sample JSON Amazon review.

We first convert this file into csv file using the dataframe.to_csv('file_name') command that is available from the panda's library [28]. Once we get the csv file, we drop the irrelevant columns like 'unixReviewTime', 'reviewerName', 'asin', 'reviewerID', etc. In the end, we keep 2 columns which are non-trivial for sentiment analysis – 'overall' and 'reviewText'. The sample review is now of the format as shown in Figure 3.2.

Overall	reviewText
5.0	I bought this for my husband who plays the piano. He is having a wonderful time playing these old hymns. The music is at times hard to read because we think the book was published for singing from more than playing from. Great purchase though!

Figure 3.2 A sample Amazon review.

3.2.2 Manually Labeled Dataset

We have used manually labeled datasets for further evaluation. Giving each review in the Amazon review dataset an emotion score that indicates whether it is positive, or negative is known as labeling. One of the critical steps in producing a labeled dataset for ML, including SA, is manually labeling the dataset, and then validating those labels.

We use the available categories of manually labeled reviews to analyze the use of FL to create high-performance SA models. The labeled datasets are of the category in Electronics review, Beauty review, and Books review data. Table 3.2 gives an overview of the manually labeled categories of review. The labeled dataset is smaller in size, and is sentence based rather than paragraph based. These categories have been manually labeled. It is a time-consuming process; hence the size of the categories are 5000 for each.

Table 3.2 Overview of Manually Labeled Categories.

Category	Positive Reviews	Negative Reviews	Total Reviews
Labeled Beauty	4,306	694	5,000
Labeled Books	2527	2473	5,000
Labeled Electronics	4686	314	5,000

In general, reviews are of average 4-8 lines, but there exists that one sentence in the review that depicts the true emotion of the customer. After manually reading the reviews and selecting that one sentence, correct labels have been assigned to that review text.

Categories	Rating	Reviews
Electronics	1	amazon kindle is always the best ebook, upgrade every new model
Beauty	0	Battery blew up in the charger
Electronics	0	It does its job but I would buy one which the screen is brighter
Books	1	Even though you already know what the outcome will be, this is still an enjoyable and erotic read
Books	0	Gave up about half way through
Electronics	0	Looking at the picture and seeing it was 8th generation I assumed it would be a great device
Books	0	I read the other reviews and decided to give it a try despite the review labeling it brutal
Electronics	1	This kindle is light and easy to use especially at the beach!!!
Beauty	1	Work Great, Great Value

Figure 3.3 A sample Manually Labeled review.

Any mistakes or discrepancies can be found and fixed, and the dataset can then be re-labeled or updated as necessary during the label validation stage. Figure 3.3 gives an example of sentence-based reviews for Books category. This dataset was manually verified using 0 for negative and 1 for positive polarity. Since errors or biases in the labels might have a negative impact on the performance of the models, validation is crucial to ensuring that the labeled data is accurate and suitable for training ML models.

3.3 Binarization of Review Ratings

Binarization is a process of converting multi-class target variable into a binary variable i.e., with 2 classes. For sentiment analysis models, binarization is required to predict whether the sentiment of the review is positive or negative [30].

We need to convert the review ratings to a binary polarity. Currently we have ratings as 1, 2, 3, 4 and 5 where 5 is considered to be very good or highly positive and 1 is very bad or highly negative. 3 is considered to be more of neutral positive. This is called 5-class classification as there are 5 classes in which a review text can be classified. We want to convert these 5-classes to 2-classes – 0 and 1. This allows us to do binary sentiment classification of reviews, which is considered to be easier for the ML model to understand [30].

The dataset mentioned in Chapter 3.2 are of 5-class classification. Hence, we convert all the ratings ranging from 3 and above as positive polarity i.e., 1 and the ratings 1 and 2 as negative polarity i.e., 0. The resulting dataset is shown in Figure 3.4.

	overall	reviewText
0	0.0	Sexyback is superwack to me. this song reminds...
1	0.0	When I first heard the title of this song, I t...
2	0.0	Terrible it won't play on my divice
3	0.0	Not worth the price: this digital download was...
4	0.0	family member downloaded for their mp3
...	...	...
169775	1.0	Casting Crowns songs are all wonderful!!
169776	1.0	Just reminds you that you are never alone.
169777	1.0	Good product, good service.
169778	1.0	I love every single song this group sings. The...
169779	1.0	Great song

Figure 3.4 Review Dataset snapshot after binarization.

The labeled dataset in Chapter 3.3.2 is manually labeled to a binary polarity. Hence, binarization process was manually applied to these datasets.

3.4 Data Preprocessing

We clean the dataset for NLP task using Natural Language Toolkit (NLTK) libraries [34]. Figure 3.5 illustrates the steps involved in data preprocessing. First, we replace any non-alphabetical and non-space characters with a space. This way we can eliminate any special characters, numbers or punctuation marks that may not contribute to the sentiment of the review. After this is achieved, we use the lower() method to convert the entire text to lower case. This method ensures that words with the same spelling, but different cases, are processed uniformly. We then split the text into individual words using split() method.

Additionally, we use the Porter stemming algorithm from the NLKT library [34] to perform stemming. Stemming is used to convert words to their root form. For example, the word “running” can be converted to its root word “run” using stemming. This process allows for the treatment of different word forms in a consistent manner, resulting in improved text analysis.

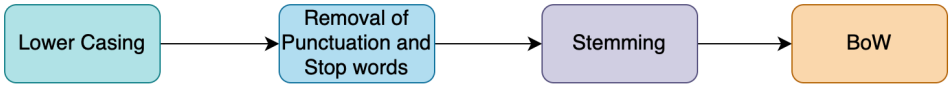

Figure 3.5 Data Preprocessing steps

Finally, we remove stop words using the set() method. Stop words like ‘and’, ‘the’, ‘a’, etc., do not contribute to the sentiment of the review. The final output is combined using the join() method from the NLKT library [34]. Now the corpus contains all clean text that is ready for analysis.

To start the analysis, we need to covert the text into numerical form. This is done using Bag of Words (BoW) model. In this method, the frequency of a word is represented using a vector. We use the CountVectorizer from scikit-learn library to make a BoW representation. Figure 3.6 shows the dataset from Chapter 3.2 which is ready for sentiment analysis.

	0	1	2	3	4	5	6	7	8	9	...	1491	1492	1493	1494	1495	1496	1497	1498	1499	y
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0.0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0.0
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0.0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0.0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
169775	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1.0
169776	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1.0
169777	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1.0
169778	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1.0
169779	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1.0

Figure 3.6 Snapshot of cleaned dataset (from Chapter 3.2).

To ensure consistency, the same pre-processing techniques were applied to the manually labeled datasets presented in Chapter 3.2.2, and the cleaned dataset is shown in Figure 3.7

	0	1	2	3	4	5	6	7	8	9	...	1491	1492	1493	1494	1495	1496	1497	1498	1499	y
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
4995	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
4996	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
4997	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
4998	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
4999	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1

Figure 3.7 Snapshot of cleaned dataset (from Chapter 3.2.2).

CHAPTER IV:

SYSTEM DESIGN

4.1 System Architecture

Using the data pre-processing techniques mentioned in Chapter III, we obtain cleaned datasets for each review category. Figure 4.1 gives an overview of how we train a sentiment analysis model for new amazon category that has no available training dataset by using labeled datasets of other categories. In Federated Learning, client refers to a node that has local data and performs model training on that local data. The local models mentioned in Figure 4.1 refers to models that train only on their respective client data. Additionally in Figure 4.1 we have a server, which is a central entity that is responsible for sending local models to the client for training and for aggregating the updated parameters received from the clients.

Each client is assigned with a review category. The server initializes weights for the local models and sends these local models to all the clients. The category of review is local to the client and is only used to train the local models. Once all the local models are trained on their respective category, each local model sends a copy of its parameters to the server.

In Figure 4.1, the server receives the model parameters from all the clients. These parameters are aggregated using a Federated Learning algorithm. The aggregation of all the local model parameters into a single model builds a new and improved model, which is known as the global model. Now, the global model has the features of all the categories without having the need to access each category as only the local models train from them. In this process, the Federated Learning approach allows each category to remain local to its client and still contribute to the global model's training.

The new global model is sent to the clients as local models for further training. From Figure 4.1, the server sends the global model to the clients for next round of training. This process is repeated for a reasonable number of iterations. The number of iterations depend on the size of the dataset. For larger datasets, a greater number of iterations are required to converge. In Machine Learning, a model converges when it has achieved the best possible values for the model parameters. Whereas, if the computational resources are limited, smaller number of iterations are preferred to avoid exhausting these resources. Once all the iterations are completed, we have a global model that can be used to predict the sentiments of category without labeled dataset of review.

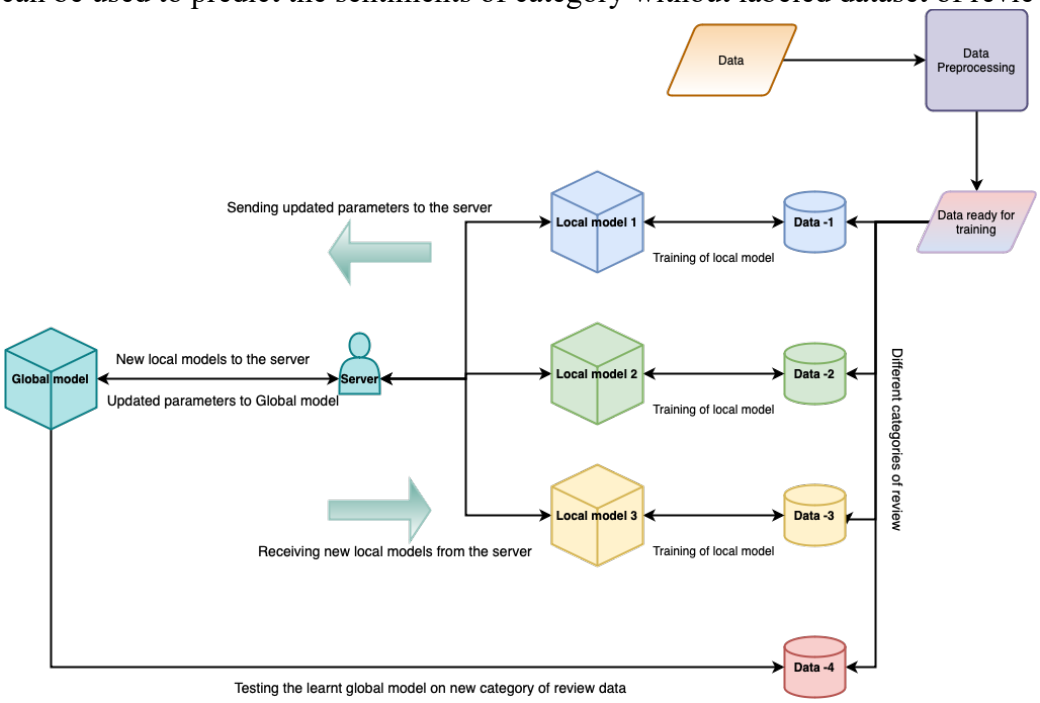

Figure 4.1 System design.

4.2 Local Model and Global Model Training

4.2.1 Local Model

Every client, or in our case every category of review has a local model which is just used to train from its category. For this local model, we are using a neural network. The neural network is a Multi-Layer Perceptron (MLP) which has 3 layers. Each neuron

in the input layer is a representation of a features of the input data. In sentiment analysis model, features are the presence of certain words in the review texts, the length of the review text or the part of speech of the certain words. NLTK Libraries [34] can be used to automatically identify the most relevant words or phrases based on their frequency. This approach is known as feature selection. For example, if we consider a review for a guitar, NLTK library looks for words like "sound quality," "playability," "tone," "neck," "fretboard," "pickup," "price," and "brand," in the review text among others. The size of the input layer is decided based on the size of the X_train vector shape. Hence it is important for the size of the input layer to be equal to the number of features of the input data. The output of this neural network gives the sentiment of the review text, which is either 0 or 1. For the Multi-layer perceptron in Figure 4.2, the input layer has 1500 neurons, the hidden layer has 15 neurons, and the output layer has a single neuron.

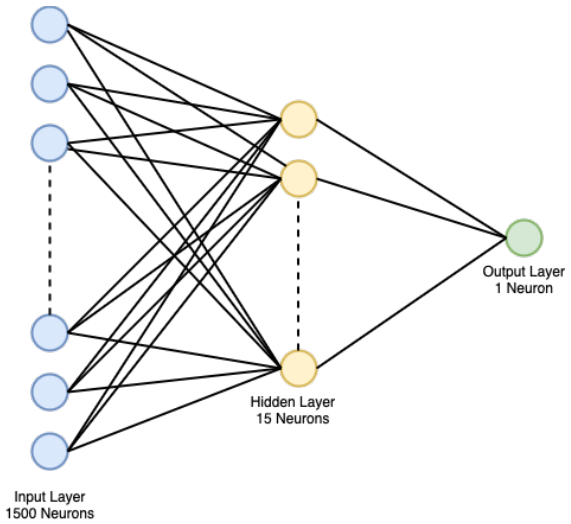

Figure 4.2 MLP - Neural Network.

We use the ReLu activation for the input, the hidden layer and sigmoid activation for the output layer.

Client Model Optimizer Function

The function client_model_optimizer is called inside the Federated Learning training loop. This function is used to create a copy of the global model and sends the copy of this model to the clients involved in the training. This function has two arguments as input – client which is the specific client that receives the model copy and the global model that will be copied. The copy of the global model is created using the copy() keyword and this copy is sent to the client using the send() keyword from the PyTorch library [33]. After the copy is sent to the client, this function initializes the gradient descent used for training. The gradient is used to minimize the loss function of a ML model. A loss function is the measure of the difference between the predicted value and the actual value. We aim to reduce this difference in order to make accurate predictions. We use the Stochastic Gradient Descent (SGD) with a learning rate of 0.2. The value of learning rate gives the rate by which the model parameters are updated in each iteration. Choosing a high learning rate can result in an overshoot of optimal solution and diverge [31]. The optimization process may not converge to the minimum point of the loss function. The loss may start increasing again, which leads to poor model performance. This function returns the local copy of the global model and the gradients used for local training on the client node.

Training Each Client Function

This function is used to train each client's model on its local data. This function takes the following arguments – the optimizer for the local model, the local model, data of the client and the target values for the input data. This function returns the updated client model, the loss value, and the gradients. This function is implemented in the following steps.

1. The gradients of the parameters of the local model with respect to the loss function are initialized to 0. If the gradients are not cleared at the start, it may cause unexpected behavior. This is because PyTorch stores gradients on every backward pass [33].
2. The local data is passed over to the local model to generate the outputs.
3. The loss function is applied to the predicted output and the target labels. This process is used to calculate the loss value.
4. The gradients of the loss value are calculated with respect to the parameters of the model using back propagation.
5. The optimizer uses the gradients calculated in the previous step to update the model's parameters.

4.2.2 FedAvg Algorithm

The most commonly used algorithm in federated learning for text-based analysis is FedAvg. As the name suggests, this algorithm averages the weights obtained by the client and sends the aggregated weights to the server. This algorithm in Figure 4.2 was introduced by McMahan et al. and describes the practical implementation of Federated Learning [8].

FedAvg , is an extended implementation of Federated Stochastic Gradient Descent (FedSGD [7]). For fixed learning rate η and the total number of clients K, the pseudocode for the FedAvg algorithm is given in Figure 4.2 [8].

1. The server initializes the model parameters (w_0) at the start of training.
2. For each round of training (t = 1,2,...), the server selects a random subset of clients (S_t) to participate in the training process based on the value of *m*.

   $S_t \leftarrow$ (random set of *m* clients)

   The number of clients *m* is selected based on a maximum value ($C \cdot K$, 1).

The hyperparameter C controls the fraction of clients selected in each round of training, and its value is determined through a process of trial and error. C can take values ranging from 1 to K, and the value that results in the best performance is selected. This approach aims to optimize the performance of the FedAvg algorithm by selecting the optimal fraction of clients to participate in each round of training.

3. Each selected client (k) performs a local update of the model parameters (w_t) in parallel with other clients using the ClientUpdate function.

$$w_{t+1}^{k} \leftarrow ClientUpdate(k, w_t)$$

The ClientUpdate function is a local update function performed on each client. The client splits its local data (P_k) into batches of size B and performs multiple epochs (E) of training on each batch (b) using gradient descent (∇). The weights are updated by subtracting the product of the learning rate (η) and the gradient of loss function (l) from the current weights.

$$w \leftarrow w - \eta \nabla l(w; b)$$

4. After each epoch, the updated model parameters (w) are returned to the server.
5. The updated model parameters from each client are averaged and then the averaged value is assigned to the new global model parameter (w_{t+1}).

$$w_{t+1} \leftarrow \sum_{k \in St} \frac{n_k}{m_k} w_{t+1}^{k}$$

where m_k is the aggregated value of n_k for the selected clients at each round of the Federated Learning algorithm [8].

Algorithm 1 `FederatedAveraging`. The K clients are indexed by k; B is the local minibatch size, E is the number of local epochs, and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
$m \leftarrow \max(C \cdot K, 1)$
$S_t \leftarrow$ (random set of m clients)
for each client $k \in S_t$ **in parallel do**
$w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
$m_t \leftarrow \sum_{k \in S_t} n_k$
$w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{m_t} w_{t+1}^k$ *// Erratum*[4]

ClientUpdate(k, w): *// Run on client k*
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
for batch $b \in \mathcal{B}$ **do**
$w \leftarrow w - \eta \nabla \ell(w; b)$
return w to server

Figure 4.3 FedAvg Algorithm [8]

4.2.3 Global Model

The global model in federated sentiment analysis is a Machine Learning model that is built by combining the local model parameters from the participating clients. The goal of the global model is to be able to predict the sentiments of a category of review which does not have a labeled training data. The local models' parameters are pooled after each round of training to create the global model, which is typically trained over several rounds. The performance of the global model is enhanced over time by this repeated process of training and aggregation.

The important steps involved in the training of the global model are shown in Figure 4.4. The function used to train the global model takes the following inputs – all the local models and global model. Firstly, we find the number of local models. We initialize variables to store the aggerated weights.

In a loop, we use the get() method as shown in Figure 4.4 to get the local model's parameters to the central server. These parameters from all the local models are summed

and stored in the corresponding variables that were initialized in the beginning. These variables now store the sum of the local model's parameters.

Then the sum of the parameters are averaged by dividing them with the number of clients. As shown in Figure 4.4, the set() function is used replace the parameters of the global model with these new averaged parameters. At the end, the new updated global model is returned.

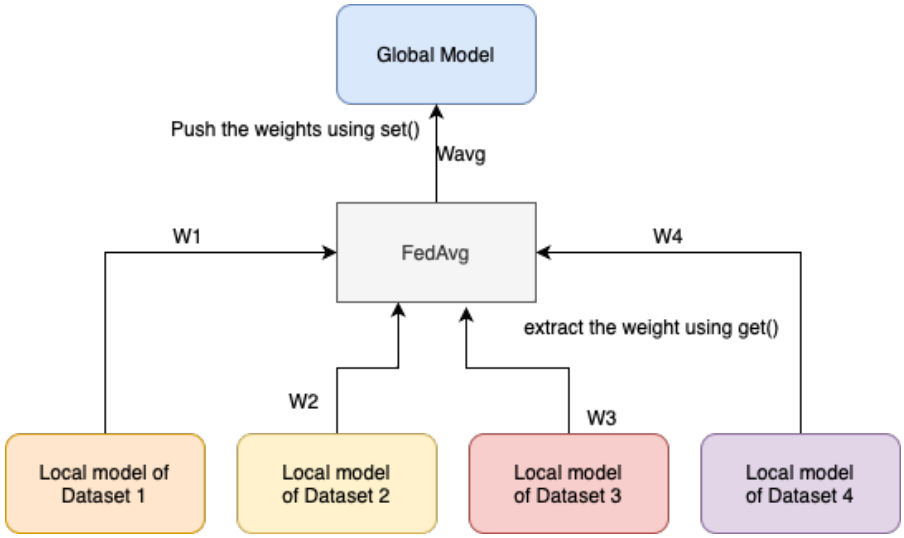

Figure 4.4 Updating the Global Model Parameter.

4.3 Federated Learning Implementation

4.3.1 PySyft Libraries for Federated Learning

We make use of the OpenMind's PySyft libraries to implement the Federated Learning models mentioned in Chapter 4.2. It is an open-source library which has various methods to perform private Machine Learning functions. PySyft provides tools to execute distributed Machine Learning tasks and Federated Learning. This library allows collaboration of multiple datasets to perform Machine Learning operations without the need of sharing the data with each other or with an authority.

4.3.2 Client Initialization

In order to start our implementation, we need to set up our Federated Learning environment. In our study, our aim is to train the global model with four different review categories and test it with a completely different category. This implies that we need four different clients and one server. Each client will have one category of review as its local data and as there are four categories of reviews, each category will be assigned to a client. Hence, we have four clients.

PySyft libraries provides virtual workers which is used to create workers – clients or server for Federated Learning. We can access the virtual workers by initializing a hook. This hook object is used to add the PySyft functionality into PyTorch framework [33]. Once the hook object is initialized, we can access the virtual workers by using the sy.VirtualWorker() function. This function takes the hook object and the client ID which can be defined by you and can be a string as an input. For example, for client 1 the virtual worker will be defined as following –

```
client1 = sy.VirtualWorker(hook, id='client_1').
```

4.3.3 Federated Dataset

To train a machine learning model on a distributed dataset using PySyft, we need to create a federated dataset. The federated dataset contains a reference to the individual category but the data itself remains on the respective client node. The individual categories are not combined to form a single dataset. The local categories of review are the individual categories and the reference to these categories are stored in the federated dataset. The reference to each individual category in the federated dataset contains information about the location and access permissions of each category.

PySyft provides a function that can take the individual client dataset as input and create a Federated Dataset. This federated dataset is now given to a FederatedDataLoader

which will iterate over remote batches. FederatedDataLoader is a function from PySyft to create a data loader. Unlike centralized dataset, we need an interface where the data is only accessed by the client and should also be able to create batches of individual client data. This cannot be done by a traditional data loader as the data is private to each client.

4.3.4 Federated Training

The federated training is performed by invoking a training function. The training functions takes the following arguments as input - list of all clients, model, federated data loader, validation data, local data loader, number of epochs, C, and local iterations which is the number of iterations per client per round.

C is the number of clients that will be selected randomly for each iteration to perform local training on their respective categories of review. The selected clients will be used to train their models locally and send the updated models to the central server for averaging. The client model optimizer function is called to send out a copy of this model back to its client and also to get the optimizer for that client model.

Below are the steps involved in training the global model –

1. The number of epochs are set to 0 to start the training.
2. For every round of training, C clients are selected for training.
3. For every client in the subset C, local model is created and sent to the client using the client_model_optimizer function mentioned in Chapter 4.2.1. This function returns the local model and gradients of the local model. These two variables are stored in lists.
4. Using the train_each_client function from Chapter 4.2.1, each local model is trained for specific local iterations on the federated data loader. The lists storing the local models and gradients of the local models are updated again.

5. Once all the clients from the subset C have been trained, the global model is trained using the function described in Chapter 4.2.3. The global model receives the aggregated parameters using this function and these parameters are stored in the model variable.
6. All the above steps are repeated until the number of epochs are completed.

Once all the epochs are completed, we have a new and improved model – global model that is built using the local model's parameters.

CHAPTER V:

EXPERIMENTS AND RESULTS

5.1 Experimental Setup

We aim to develop a model that can be used to predict the sentiments of a review category without labeled dataset in case of unavailability of labeled training data. Hence, to test how our approach performs, we need to test the global model's performance against a new review category. We use the AUPRC score to understand the performance of the global model on a different domain of review dataset.

For evaluating the Federated Sentiment Analysis (FSA), we take multiple scenarios under consideration. The main focus of all the experiments is to understand and analyze the Federated Sentiment Analysis model's performance when there is no availability of trained labeled category. Hence in all approaches, the model is tested on a different category of data whose language and context of reviews remains unknown to the global model. Hence, we consider the following experiments.

5.1.1 Federated Sentiment Analysis Model Trained using 4 Different Categories

We use four different categories as shown in Figure 5.1 - Industry & Science, Software, Musical Instruments and Digital Music to train the global model by utilizing the Federated Learning environment and compare it to a model using a centralized dataset (The four categories have been combined into a single dataset). The main focus of our thesis is to analyze how the global model performs on a category without labeled dataset when it has been trained on different categories. The category without labeled dataset is the beauty category as this category has not been used for model training. We compare these results with the traditional way of doing sentiment analysis, also known as the centralized sentiment analysis. This experiment provides an insight to our first research question.

We want to build a global model that fits and accurately predicts the sentiments of variety of review categories even if the language and context of the new review category is unknown to the model. In the field of Machine Learning, it is common to train a model on one dataset and test it on another to check and verify the model's performance on fresh, untested dataset. The language and context of each review category depends on the user, the product, and the domain. It is important to train the model on a set of review data and then test it on a completely different review data to understand the capabilities of the built model.

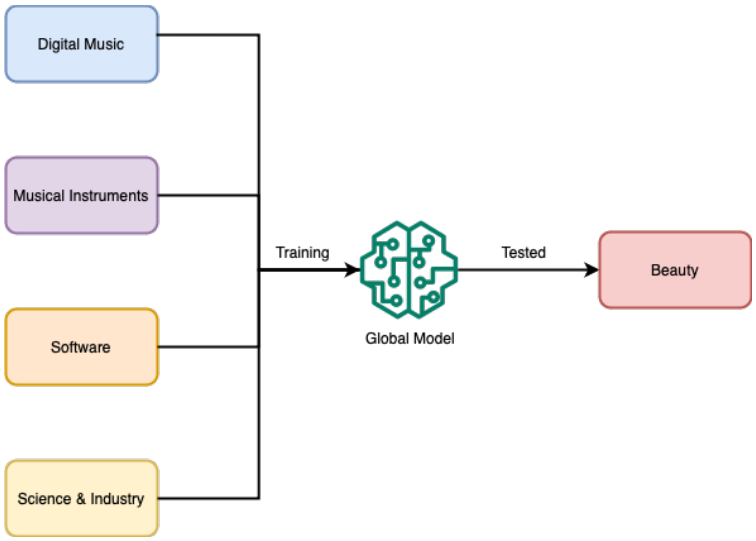

Figure 5.1 Global model trained using 4 different categories.

5.1.2 Federated Sentiment Analysis Model Trained using Small Dataset

This experiment aims to investigate the second research question**.** This approach allows us to understand the impact of smaller dataset size on the performance of the global model**.** From Figure 5.2, a small subset of the Beauty category dataset with a size of 5000 is used for training the global model, which is subsequently evaluated on a larger dataset of the same category with a size of 20,000.

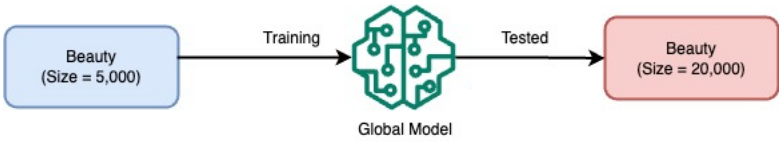

Figure 5.2 Global model trained using small dataset.

This is compared with traditional sentiment analysis, where we use the same Beauty category with a size of 5000 to train the model and test on the same category with a size of 20,000. We analyze how Federated Learning performs compared to a traditional sentiment analysis in training on a small dataset and testing on a larger dataset of the same category. By comparing these two approaches, we can determine which method is more effective in this particular scenario.

5.1.3 Federated Sentiment Analysis Model Trained using Combinations of Categories

This experiment is used to analyze the impact of the number of categories to the performance of the Federated Sentiment Analysis models. This experiment aims to analyze the impact of adding a new category of training data on the performance of the global model. We investigate the number of categories and type of categories used for global model training and their effect on the performance of global model.

For this, we use different combinations of the datasets – single dataset, combinations of two datasets and combinations of three datasets to train the global model and test it out on the unknown dataset. As we have four training categories, we test out all the possible combinations starting from a single category, and then we test two categories and then three categories. The datasets mentioned in Chapter 3.2.1 – Software, Digital music, Industry & Science and Musical instruments are used for training the global model. The model is then tested on the Beauty category.

From Figure 5.3, we have three training and testing setups - First for single category, second for combinations of two categories and third for combinations of three categories.

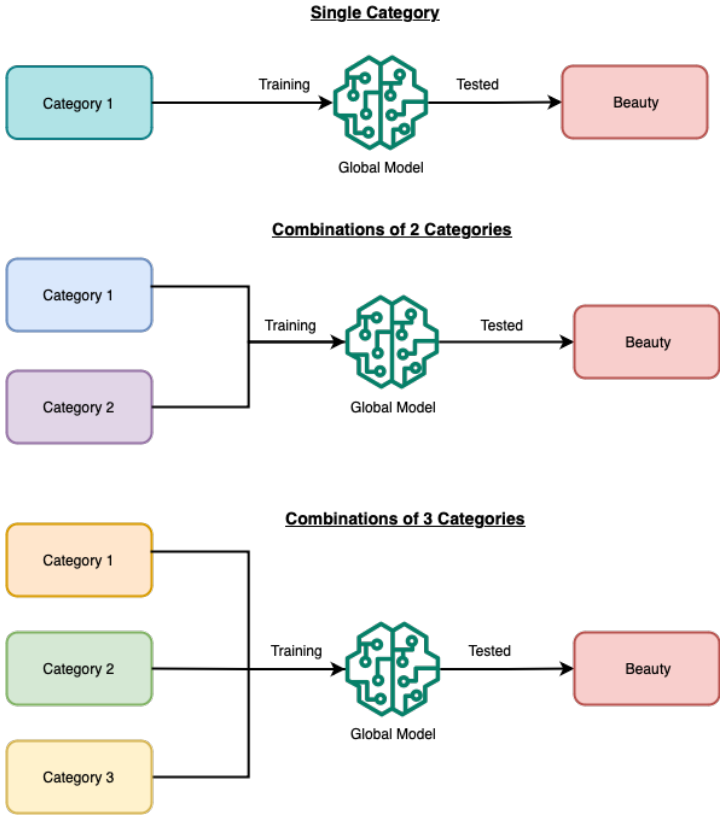

Figure 5.3 Global model trained using different combinations of categories.

For single datasets, we train the global model on say “Software” category and test it out on the Beauty category. Training a global model on a single category dataset can provide valuable insights into the model's behavior in the presence of limited training data. After we use this process for all the categories, we move on to using combination of two categories like "Software" and "Musical Instruments." We train the global model on all the possible combinations of two categories and test all the models on Beauty category. Additionally, we make combinations of three categories, for instance – “Science & Industry”, "Software" and "Musical Instruments" and test the model on Beauty category. We repeat this for all possible combinations of three categories. We

compare all the combinations of single datasets, two datasets and three datasets to analyze the impact of category type and number of categories on model's ability to label a category without labeled dataset.

5.1.4 Federated Sentiment Analysis Model Trained using Diverse Set

This experiment helps us analyze the performance of the global model when the data is more distributed among the clients. We use the datasets from Chapter 3.2.1 – Software, Industry & Science, Digital Music, and Musical Instruments for training. We combine all four categories of review dataset, shuffle them, and then randomly form subsets. These subsets are the new distribution. The size of each subset is 26,666 as the combined dataset is equally divided into three sets. As all the categories are equally distributed, every client has characteristics of different categories for its local model's training.

From Figure 5.4, these subsets are then used to train the global model and test it again on a completely different category. The number of subsets used to train the global model are changed to see the significance of number of subsets on the global model's performance.

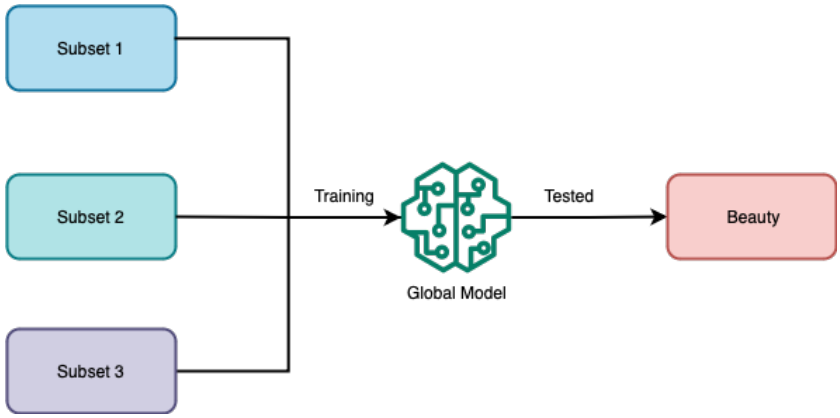

Figure 5.4 Global model trained using subsets.

5.1.5 Federated Sentiment Analysis Model Trained using Different types of Datasets

In this experiment, we investigate the performance of Federated Sentiment Analysis model on different types of review datasets: sentence-based and paragraph-based datasets. The goal is to compare the performance of model on sentence-based and paragraph-based categories. This experiment seeks to determine whether the granularity of the review text has any significant effect on the accuracy of the sentiment analysis model.

The sentence-based dataset used in this experiment has been manually labeled as mentioned in Chapter 3. For this approach, we have three categories of reviews, and these three categories are of two sub-types. We have the manually labeled sentence-based categories of Electronics, Books and Beauty from Chapter 3.2.2, and we have the labeled paragraph-based categories of Electronics, Books and Beauty from Chapter 3.2.1. We compare the performance of global model in two scenarios as shown in Figure 5.5.

1. Global model trained on sentence-based manually labeled Electronics and Books and tested on sentence-based manually labeled Beauty category.
2. Global model trained on paragraph-based labeled Electronics and Books and tested on paragraph-based labeled Beauty category.

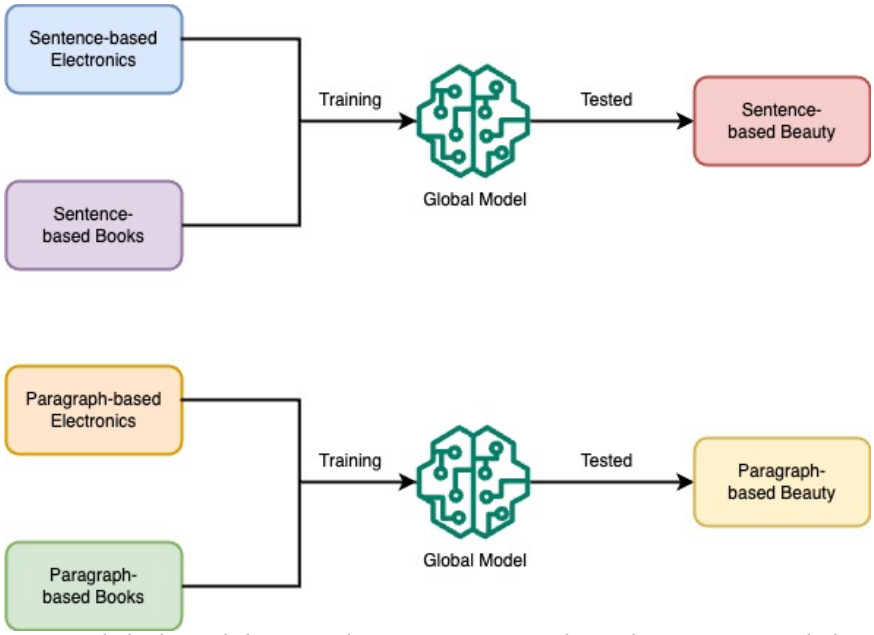

Figure 5.5 Global model trained using Sentence-based vs Paragraph-based.

We compare the model trained using manually labeled sentence-based dataset to with paragraph-based dataset to understand the importance of the type of dataset.

5.2 Evaluation Techniques

To understand the model performance, we print the Area Under the Precision-Recall Curve (AUPRC) score. AUPRC score is a more suitable metric than accuracy in imbalanced datasets where the classes are not equally represented. In such cases, accuracy can be misleading because it gives equal weight to both classes [32].

The dataset is split into 3 sets: training set which is used to train the model, validation set which is used to fine-tune the model's parameters and testing set which is used to assess the model's performance. The testing set is Beauty category, and the training set are the different review categories mentioned in Chapter 5.1. The validation set consists of 25% of the training set, which is used for cross-validation. This process helps ensure that the model is not overfitting to the training data and can generalize well to new data.

5.3 Results and Discussion

5.3.1 Results for Federated Sentiment Analysis Model Trained using Four Different Categories

These results can be used to answer our research question i.e., does the global model perform better than a model trained using traditional sentiment analysis approach? We build a global model from different categories to further predict the sentiments of a category without labeled dataset. These results indicate that we can use Federated Learning to build a global model from different categories of reviews which can predict the labels on a category without labeled dataset.

We compare the Federated Learning approach to the centralized sentiment analysis approach. All the four categories of review are combined into one dataset to perform centralized sentiment analysis. To understand the performance of the federated approach better, we use MLP from Chapter 4.2.1 to analyze the sentiments of the reviews in the traditional environment.

We also investigate the behavior of the global model when the size of the dataset is increased. It is important to know if larger datasets improve the model's performance or if the additional data leads to diminishing results. From Figure 5.6, the AUPRC score of 0.8 was reported in the traditional approach. We have two scenarios; we use a dataset size of 10,000 for the four different categories. The global model trained on this set of reviews achieved an AUPRC score of 0.8486. From this AUPRC score we understand that the model performs reasonably well at classifying the positive and negative sentiment in reviews.

The second scenario has an increased dataset size of 20,000 for the four different datasets. This model achieved an AUPRC score of 0.887, which is higher than the

previous score. This indicates that the larger dataset leads to better model performance. This is because a larger dataset provides more training examples.

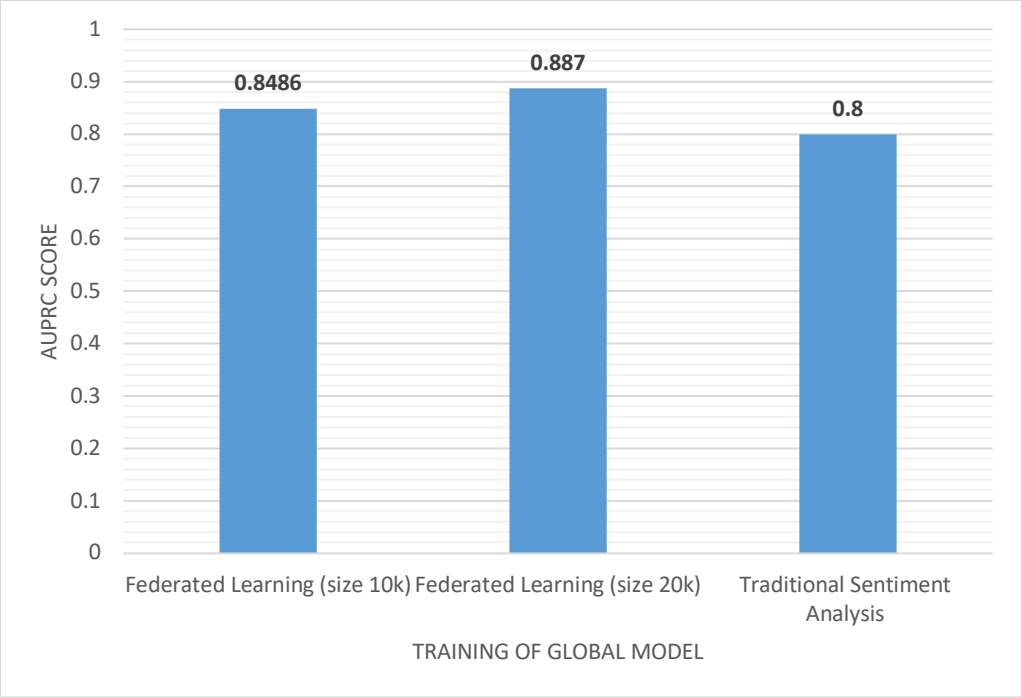

Figure 5.6 AUPRC Scores for federated learning vs traditional SA.

5.3.2 Results for Federated Sentiment Analysis Model Trained using Small Dataset

We investigate the performance of Federated Learning in comparison to traditional sentiment analysis when training on a small dataset and testing on a larger dataset of the same category. By comparing these two approaches, we aim to assess the effectiveness of each method. We can determine the potential of Federated Learning as an alternative approach for training better data analytical models for categories with no large-scale labeled datasets.

From Figure 5.7, the results show that the global model trained using Federated Learning outperforms the traditional sentiment analysis model with an AUPRC score of 0.81 compared to 0.774 for the traditional model. This indicates that Federated Learning

is a more effective approach for training models using a small dataset, which can be useful in scenarios where there is limited availability of data.

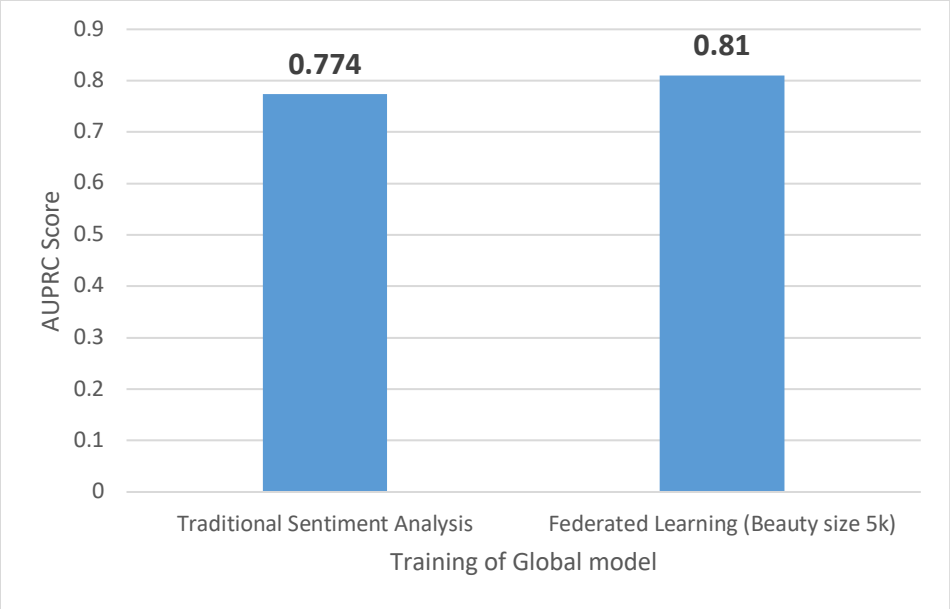

Figure 5.7 AUPRC Scores for small size datasets.

5.3.3 Results for Federated Sentiment Analysis Model Trained using Combinations of Categories

In this experimental setup, we try to figure out if the choice of categories effects the testing of the global model. We analyze this scenario by training the global model with single categories and test it out on a separate Beauty category. S denotes Software, MI denotes Musical Instruments, DM denotes Digital Music and IS denotes Industry & Science. From Figure 5.8, the AUPRC scores is the highest - 0.882 for the Musical Instruments. This score suggests that the model performed well in classifying positive and negative reviews for the testing category.

We train the model on training Beauty dataset and test it on the same category testing dataset. This model performs better with an AUPRC score of 0.9718. This is because the model is familiar with the word vocabulary and the domain of the review.

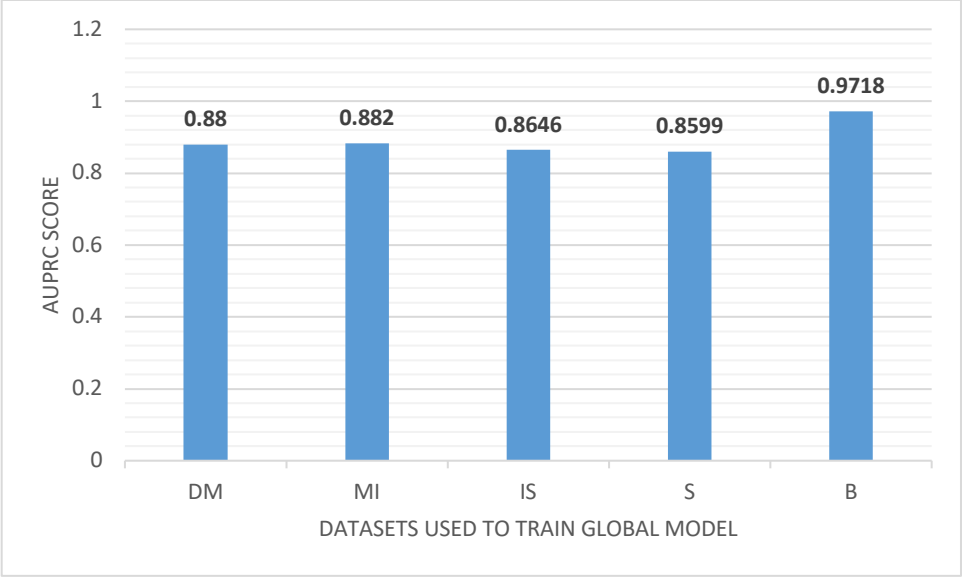

Figure 5.8 AUPRC Scores for single training dataset.

Now we consider combination of two categories. This way we understand if selecting any two categories improves the global model's performance. From Figure 5.9, we have all the possible combinations for the four datasets. We see that the Software and Digital Music dataset achieves an AUPRC score of 0.897, which is higher than the performance of the model trained on a single category -Musical Instruments. This suggests that combining these two categories improved the performance of the global model in detecting positive reviews. Digital Music and Musical Instruments have an AUPRC score of 0.877, which indicates that these two categories might have more relevance to the testing category. Hence, the model trained from these two categories performs better in predicting the sentiments of Beauty category.

The combination of Software and Digital Music datasets proved to be the most effective, while Industry & Science and Digital Music datasets resulted in the lowest AUPRC score.

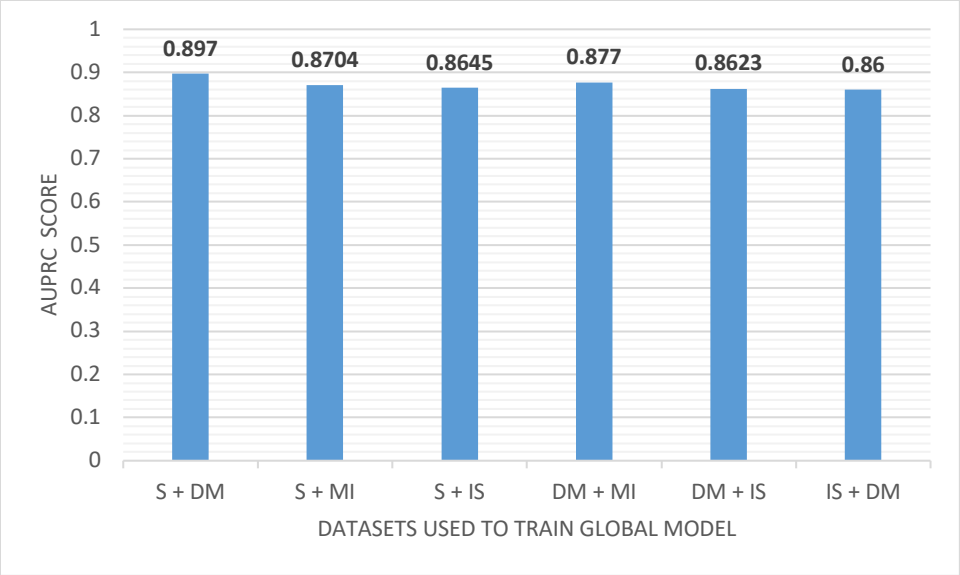

Figure 5.9 AUPRC Scores for 2 different combinations of training data.

We test the performance of the global model when we use three categories to train the model. S denotes Software, MI denotes Musical Instruments, DM denotes Digital Music and IS denotes Industry & Science. In Figure 5.10, the model trained with Software, Digital Music and Musical Instruments performed relatively high with an AUPRC score 0.8715, indicating that there may be few similarities in the sentiments of reviews across these categories. The scores for combinations of Software and Industry and Science, and Digital Music are also relatively high – 0.8702, indicating that sentiment analysis can be effective across different domains of review datasets.

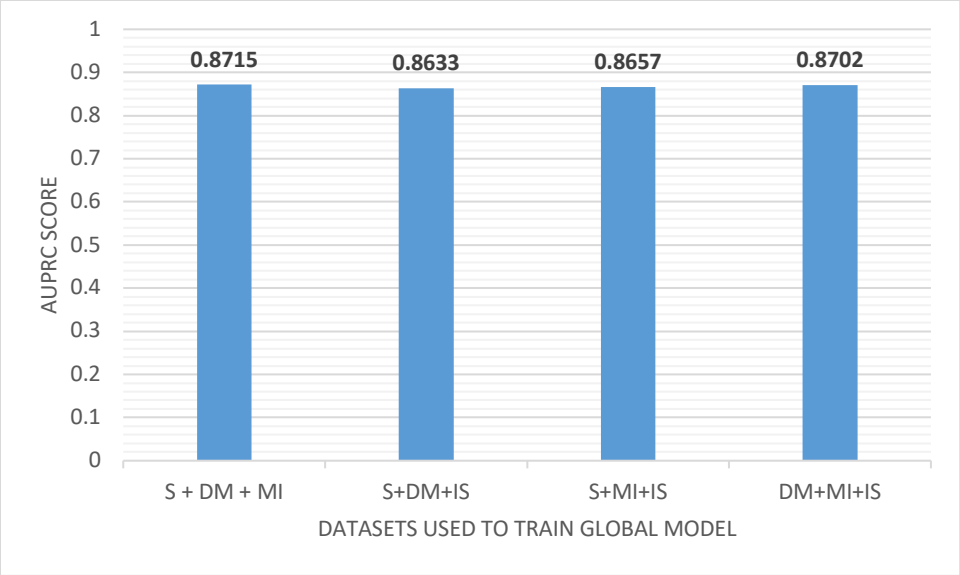

Figure 5.10 AUPRC Scores for 3 different combinations of training data.

Overall, in this experiment we see the impact on the global model's performance with respect to the number of categories and the combination of categories used for training. In our case, using two combinations of categories – Software and Digital Music gives a good sentiment analysis model to label the Beauty category when compared to a single category or three categories. These two categories might have similar word vocabulary, dialects, and review contexts with respect to our testing category of review. Hence, it is important to select the categories for training a global model depending on the requirement of the application or study.

5.3.4 Results for Federated Sentiment Analysis Model Trained using Diverse Set

The experiments from Chapter 5.1.3 is related to this experiment as both of these investigate the impact of different factors on the performance of the global model. In Chapter 5.1.3, the impact of the number and type of categories used for training is analyzed, while in this experiment, the impact of the distribution of the data among the clients is studied. Both experiments are designed to improve the performance of the global model.

In this experiment, the review datasets of four categories – Software, Industry & Science, Musical Instruments and Digital Music are combined and then randomized using a shuffling process to form three subsets of equal size. These subsets are denoted as subset 1, subset 2, subset 3, and are considered as a new distribution used to train the global model. Here, we try to analyze if creating random subsets from combining different categories improve the performance of our approach. From Figure 5.11, an AUPRC score of 0.89 was obtained when the global model was trained using all three subsets. The AUPRC score decreased to 0.8774 when subsets 1 and 2 were used for training. The AUPRC score was 0.8832 when subsets 1 and 3 were used for training, and it was 0.8726 when subsets 2 and 3 were used.

The best result was achieved when all three subsets were used, followed by subsets 1 and 3 or subsets 1 and 2. The least performance was obtained while using subsets 2 and 3.

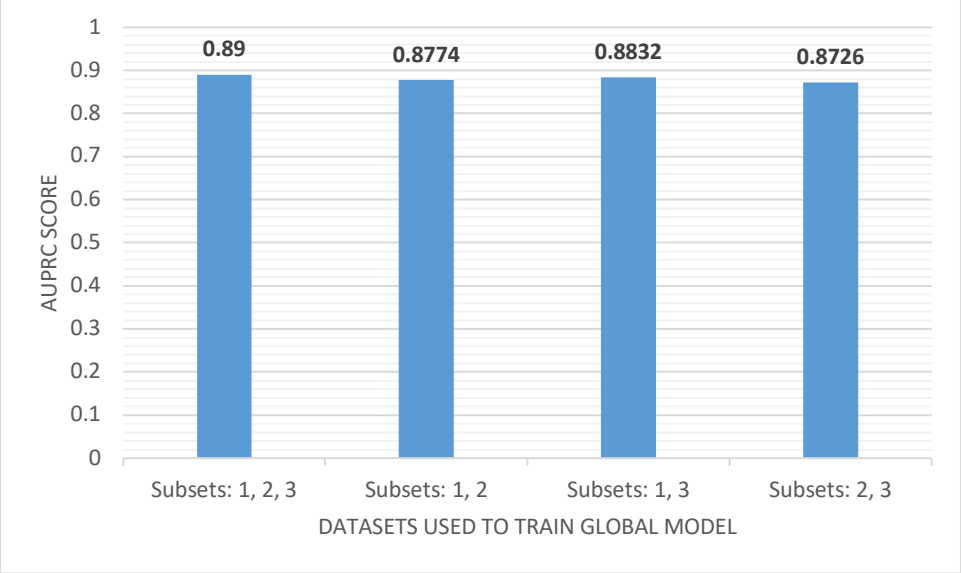

Figure 5.11 AUPRC Scores for different combinations subsets of training data.

As a whole, this experiment helps us understand that when the data is more distributed among clients, it improves the global model's performance. The observed improvement can be attributed to the fact that the local models are trained on a wider range of review contexts and vocabulary, which in turn enhances the performance of the global model as it relies on the training of the local models.

5.3.5 Results for Federated Sentiment Analysis Model Trained using Different types of Datasets

This experiment is used to analyze the performance of Federated Sentiment Analysis model on different type of datasets. We compare sentence-based review dataset with paragraph-based review dataset. We aim to investigate the impact of the review text granularity on the performance of a federated sentiment analysis model.

We consider two scenarios. Firstly, we train the global model on manually labeled sentence-based Books and Electronics category and test on manually labeled sentence-based Beauty category. From Figure 5.12, we get an AUPRC score of 0.88. Next, we train the global model with paragraph-based Books and Electronics categories and test it on paragraph-based Beauty category. We get a better AUPRC score of 0.911. One reason could be the smaller size of the sentence-based dataset, compared with the paragraph-based dataset. Overall, Federated Learning can be used in case of sentence-based datasets and paragraph-based datasets as it performs well in both scenarios.

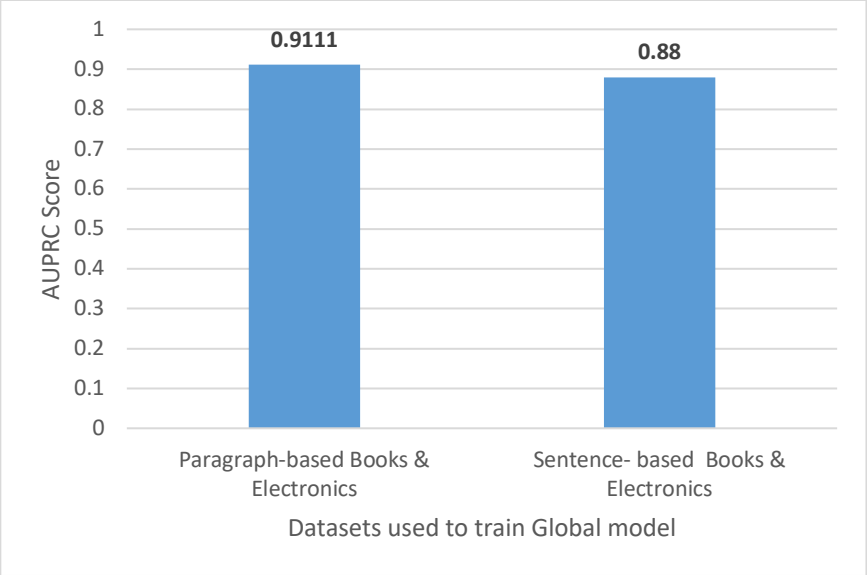

Figure 5.12 AUPRC Scores for Sentence-based vs Paragraph-Based dataset.

CHAPTER VI:

FUTURE SCOPE AND CHALLENGES

Federated Sentiment Analysis shows great promise in improving the model's performance to label a new category of data, while there are several challenges that must be addressed to fully realize its potential. One of the challenges of Federated Sentiment Analysis is selecting the appropriate categories of review for training the model. Selecting categories requires careful consideration of characteristics of the target category and the available labeled category. Future research could investigate methods for selecting datasets that are representative of the target category, while also ensuring that the resulting model is not biased towards specific categories that are overrepresented in the training data.

Another challenge is training the model on large, labeled datasets. Federated Learning approach has shown to be effective on small datasets, while its impact on large size datasets is still unclear. Future research could investigate how Federated Learning approach can be adapted to work with larger datasets. Additionally, handling imbalanced datasets is a significant challenge of Federated Sentiment Analysis. Imbalanced datasets can lead to bias that makes it difficult for the model to accurately predict the sentiments of underrepresented categories. Future research could investigate techniques for handling imbalanced datasets in a federated setting by utilizing techniques to increase the amount of labeled data for underrepresented categories.

Semi-supervised learning algorithms rely on both labeled and unlabeled data to train models, and in a federated setting, the labeled data is limited. Semi-supervised learning can be used to leverage the vast amounts of unlabeled data that are available for sentiment analysis, potentially improving the accuracy of the resulting models. Exploring semi-supervised learning is another direction for future research in Federated Learning.

CHAPTER VII:

CONCLUSION

In conclusion, this thesis proposes a Federated Learning based approach for sentiment analysis, which addresses the challenges of centralized models such as data heterogeneity, bias, and the requirement of large computational resources. The proposed approach uses multiple clients to train local sentiment analysis models on labeled review datasets of specific categories. FedAvg algorithm is used to aggregate the parameters from these models to build a global model for a new category with no labeled dataset. By evaluating the performance of this approach using Amazon review datasets, it is found that the Federated Learning-based sentiment analysis outperforms centralized sentiment analysis by 10%. This result demonstrates the potential of federated learning in training accurate and reliable sentiment analysis models for categories without labeled training datasets. Federated Learning adaptation for training models with smaller datasets led to improved performance compared to models trained using traditional sentiment analysis with smaller datasets. Selecting the right number and combination of categories for the global model training is crucial, and more distributed data among clients improves the model's performance. Overall, this research provides valuable insights and a new direction for developing more efficient and effective sentiment analysis models using federated learning.

REFERENCES

[1] Pang, Bo, and Lillian Lee. "Thumbs up? Sentiment classification using machine learning techniques." Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics, 2002.

[2] Fang, X., Zhan, J. Sentiment analysis using product review data. *Journal of Big Data* **2**, 5 (2015). https://doi.org/10.1186/s40537-015-0015-2

[3] Pankaj, P. Pandey, Muskan, and N. Soni, "Sentiment Analysis on Customer Feedback Data: Amazon Product Reviews," *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, Faridabad, India, 2019, pp. 320-322, doi: 10.1109/COMITCon.2019.8862258.

[4] Liu, M., Ho, S., Wang, M., Gao, L., Jin, Y., & Zhang, H. (2021). Federated Learning Meets Natural Language Processing: A Survey. *ArXiv, abs/2107.12603*.

[5] Li, X. C., Li, L., Zhan, D. C., Shao, Y., Li, B., & Song, S. (2021). Preliminary steps towards federated sentiment classification. arXiv preprint arXiv:2107.11956.

[6] Guliani, D., Beaufays, F., Motta, G.: Training speech recognition models with federated learning: A quality/cost framework. arXiv preprint arXiv:2010.15965 (2020)

[7] Xinghua Zhu, Jianzong Wang, Zhenhou Hong, and Jing Xiao. 2020. Empirical Studies of Institutional Federated Learning For Natural Language Processing. In Findings of the Association for Computational Linguistic*s:* EMNLP 2020, pages 625–634, Online. Association for Computational Linguistics.

[8] McMahan, H.B., Moore, E., Ramage, D., Hampson, S., & Arcas, B.A. (2017). Communication-Efficient Learning of Deep Networks from Decentralized Data. *AISTATS*.

[9] Hilmkil, A., Callh, S., Barbieri, M., Su¨tfeld, L.R., Zec, E.L., Mogren, O.: Scaling federated learning for fine-tuning of large language models. arXiv preprint arXiv:2102.00875 (2021).

[10] Sattler,F.,Marban,A.,Rischke,R.,Samek,W.:Communication-efficient federated distillation. arXiv preprint arXiv:2012.00632 (2020).

[11] Tsankova, P., & Momcheva, G. (2020). Sentiment detection with FedMD: Federated Learning via Model Distillation.

[12] Tang, H., Lian, X., Yan, M., Zhang, C., Liu, J.: D2: Decentralized Training over Decentralized Data. arXiv e-prints arXiv:1803.07068 (2018).

[13] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977 (2019).

[14] Liu, Y., Chen, J., Xie, J., & Zhao, P. (2020). Privacy-Preserving Federated Learning for Sentiment Analysis. IEEE Transactions on Information Forensics and Security, 15, 1957-1972. DOI: 10.1109/TIFS.2020.2979683.

[15] T. Li, A. K. Sahu, A. Talwalkar and V. Smith, Federated Learning: Challenges, Methods, and Future Directions, in IEEE Signal Processing Magazine, vol. 37, no. 3, pp. 50-60, May 2020, doi: 10.1109/MSP.2020.2975749.

[16] C. Dwork and A. Roth, Privacy-Preserving Machine Learning: Threats and Solutions, published in Foundations and Trends in Theoretical Computer Science, August 2014, DOI: 10.1561/0400000060.

[17] Jatla Srikanth, Avula Damodaram, Yuvaraja Teekaraman, Ramya Kuppusamy, Amruth Ramesh Thelkar, "Sentiment Analysis on COVID-19 Twitter Data Streams Using Deep Belief Neural Networks", *Computational Intelligence and*

Neuroscience, vol. 2022, Article ID 8898100, 11 pages, 2022. https://doi.org/10.1155/2022/8898100

[18] Han Qin, Guimin Chen, Yuanhe Tian, and Yan Song. 2021, Improving Federated Learning for Aspect-based Sentiment Analysis via Topic Memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3942–3954, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[19] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in Conference on Artificial Intelligence and Statistics. 2017.

[20] Hard, A., Rao, K., Mathews, R., Beaufays, F., Augenstein, S., Eichner, H., Kid- don, C., Ramage, D.: Federated learning for mobile keyboard prediction. ArXiv abs/1811.03604 (2018).

[21] Guliani, D., Beaufays, F., Motta, G.: Training speech recognition models with federated learning: A quality/cost framework. arXiv preprint arXiv:2010.15965

[22] Smith, V., Chiang, C.K., Sanjabi, M., Talwalkar, A.: Federated Multi-Task Learning. arXiv e-prints arXiv:1705.10467 (2017).

[23] Caldas, S., Meher Karthik Duddu, S., Wu, P., Li, T., Koneˇcn´y, J., McMahan, H.B., Smith, V., Talwalkar, A.: LEAF: A Benchmark for Federated Settings. arXiv eprints arXiv:1812.01097 (2018).

[24] Fallah, A., Mokhtari, A., Ozdaglar, A.: Personalized federated learning: A meta- learning approach. arXiv preprint arXiv:2002.07948 (2020).

[25] Kairouz, P., McMahan, H.B., Avent, B., Bellet, A., Bennis, M., Bhagoji, A.N., Bonawitz, K., Charles, Z., Cormode, G., Cummings, R., et al.: Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977 (2019).

[26] H. Q. Abonizio, E. C. Paraiso and S. Barbon, "Toward Text Data Augmentation for Sentiment Analysis," in *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 5, pp. 657-668, Oct. 2022, doi: 10.1109/TAI.2021.3114390.

[27] Liang W, Chen X, Huang S, Xiong G, Yan K, Zhou X. Federal learning edge network-based sentiment analysis combating global COVID-19. Comput Commun. 2023 Apr 15;204:33-42. doi: 10.1016/j.comcom.2023.03.009. Epub 2023 Mar 22. PMID: 36970130; PMCID: PMC10030440.

[28] Anon (2022) Focus: Pandas: The Popular Python Library for Data Analysis and Data Science.

[29] Rosenthal, Sara, et al. SemEval-2014 Task 9: Sentiment Analysis in Twitter. 2019, https://doi.org/10.48550/arxiv.1912.02990.

[30] Joshi, Prateek. Artificial Intelligence with Python : Build Real-World Artificial Intelligence Applications with Python to Intelligently Interact with the World Around You. 1st edition, Packt, 2017.

[31] M. H. Munna, M. R. I. Rifat and A. S. M. Badrudduza, "Sentiment Analysis and Product Review Classification in E-commerce Platform," 2020 23rd International Conference on Computer and Information Technology (ICCIT), DHAKA, Bangladesh, 2020, pp. 1-6, doi: 10.1109/ICCIT51783.2020.9392710.

[32] Ling, Charles & Huang, Jin & Zhang, Harry. (2003). AUC: A Better Measure than Accuracy in Comparing Learning Algorithms. Canadian Conference on AI. 329-341. 10.1007/3-540-44886-1_25.

[33] Julian, David. Deep Learning with PyTorch Quick Start Guide. 1st edition, Packt Publishing, 2018

[34] Bird, S. et al. (2009) Natural Language Processing with Python. 1st edition. O'Reilly Media, Inc.

[35] Jure Leskovec and Andrej Krevl, SNAP Datasets: Stanford, Large Network Dataset Collection, http://snap.stanford.edu/data, jun, 2014.

APPENDIX A:

ACRONYMS

NLP – Natural Language Processing

FL – Federated Learning

ML – Machine Learning

FLEN - Federated Learning Edge Network

AUPRC – Area Under Precession Recall Curve

BoW – Bag of Words

TP – True Positive

TN – True Negative

FP – False Positive

FN – False Negative

FSA – Federated Sentiment Analysis

NLKT – Natural Language Toolkit

MLP – Multi-Layer Perceptron

FedMD – Federated Learning via Model Distillation

FedAvg – Federated Average

FedSGD - Federated Stochastic Gradient Descent

SGD – Stochastic Gradient Descent