REPUBLIC OF TURKEY
YILDIZ TECHNICAL UNIVERSITY
DEPARTMENT OF COMPUTER ENGINEERING

SEQUENCING OF EXAMPLES IN SEMI-SUPERVISED LEARNING

16011038 — Talha Bacak

SENIOR PROJECT

Advisor
Assoc. Prof. Mehmet Fatih Amasyali

June, 2022

ACKNOWLEDGEMENTS

I would like to thank Yıldız Technical University Computer Engineering Department for giving me the opportunity to realize this and many similar projects and all of my teachers for everything they have contributed to me throughout my education life.

I am also very grateful to have Assoc. Prof. Mehmet Fatih Amasyali as my supervisor. He provided me with the necessary sources, taught me a lot and guided me throughout the work i have done.

Talha Bacak

TABLE OF CONTENTS

LIST OF SYMBOLS v

LIST OF ABBREVIATIONS vi

LIST OF FIGURES vii

LIST OF TABLES viii

ABSTRACT ix

ÖZET xi

1 Introduction 1

2 Literature Review 2

3 Feasibility 3

3.1 Technical Feasibility . . . 3

3.1.1 Software Feasibility . . . 3

3.1.2 Hardware Feasibility . . . 3

3.2 Time Feasibility . . . 4

3.3 Economic Feasibility . . . 5

3.4 Legal Feasibility . . . 5

4 System Analysis 6

4.1 The goal of the project . . . 6

4.2 Draft Diagram . . . 6

4.3 Data Flow Diagram . . . 7

4.4 Use Case Diagram . . . 7

5 System Design 9

5.1 Software Design . . . 9

5.1.1 Prepare Data and Create Model . . . 9

5.1.2 Semi-supervised Algorithm . . . 10

5.1.3 Result . . . 10

5.2 Database Design 10
5.3 Input-Output Design 10

6 Application **11**
6.1 Self-training 11
6.2 Co-training 12
6.3 Tri-training 12
6.4 Tri-training with disagreement 13

7 Experimental Result **14**
7.1 Experimental Data Exploration 14
7.2 Supervised Train (without Unlabel Data) 14
7.3 Semi-supervised Train (with Unlabel Data) 15
7.3.1 Self-Training 16
7.3.2 Co-Training 16
7.3.3 Tri-Training 16
7.3.4 Tri-Training with Disagreement 17
7.4 Behavior of tagging unlabeled data 17
7.4.1 Data Exploration 17
7.4.2 Examining unlabeled data 18

8 Performance Analysis **21**
8.1 Comparison 21
8.1.1 Time 21
8.1.2 Memory 21
8.1.3 Training Dataset Size 21
8.1.4 Unlabel Dataset Size 22
8.1.5 Text or Image 22
8.1.6 Algorithms 22
8.2 Behavior of Tri-training with disagreement 22

9 Result **27**

References **28**

Curriculum Vitae **29**

LIST OF SYMBOLS

Π	Pi

LIST OF ABBREVIATIONS

CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Ranndom Access Memory

LIST OF FIGURES

Figure 3.1 Gantt Chart 4
Figure 3.2 Organization Chart 4
Figure 4.1 Draft Diagram 6
Figure 4.2 Data Flow Diagram 7
Figure 4.3 Use-Case Diagram 8
Figure 5.1 State diyagramı 9
Figure 5.2 For example 10
Figure 6.1 General self-training algorithm [4] 12
Figure 6.2 General co-training algorithm [5] 12
Figure 6.3 General tri-training algorithm [6] 13
Figure 7.1 Distribution of Dataset 18
Figure 7.2 Self-training 19
Figure 7.3 Co-training 19
Figure 7.4 Tri-training 19
Figure 7.5 Tri-training with disagreement 20
Figure 8.1 Tri-training with disagreement example 1 (small mnist) 23
Figure 8.2 Tri-training with disagreement example 2 (small mnist) 24
Figure 8.3 Tri-training with disagreement example 3 (mnist) 25
Figure 8.4 Tri-training with disagreement example 4 (tweets) 26

LIST OF TABLES

Table 3.1 Hardware Feasibility 3
Table 7.1 Dataset information 14
Table 7.2 20 epochs train 15
Table 7.3 200 epochs train 15
Table 7.4 Self-training results (20 epochs) 16
Table 7.5 Co-training results (20 epochs) 16
Table 7.6 Tri-training results (20 epochs) 17
Table 7.7 Tri-training with disagreement results (20 epochs) 17
Table 7.8 Dataset information 18

ABSTRACT

Sequencing of Examples in Semi-Supervised Learning

Talha Bacak

Department of Computer Engineering
Senior Project

Advisor: Assoc. Prof. Mehmet Fatih Amasyali

Today, the number of data is increasing rapidly day by day. These data are very important for deep learning models. In supervised learning, data needs to be labeled, but the data is growing so fast that this data cannot be labeled enough. For this, the importance of semi-supervised algorithms is increasing. How successful these algorithms are in which situations is also starting to gain importance. In this study, we examined self-training, co-training, tri-training and tri-training with disagreement algorithms. The effects of the unlabeled data size of these algorithms and the size of the training data are also examined. A library has been written for these algorithms. Tests have been made with this library. Studies were carried out for both text classification and image classification with 5 separate data sets. As a result of the studies, as the size of the training data increased, an increase in success was observed in general. It has been observed that the tri-training and tri-training with disagreement algorithms are greatly affected by the unlabeled data size, and the training times of these 2 algorithms have taken a long time. The tri-training algorithm did not generally provide an increase in success. Tri-training with disagreement algorithm has increased success under favorable conditions. Although self-training and co-training algorithms did not adversely affect success in any case, they did not increase success in all cases. According to the data sets, the situation of increasing the success has changed a lot. As the situations where these algorithms and other semi-supervised algorithms are affected are determined, the most suitable algorithms will be in a state where they can be optimized under the most suitable conditions. It will be important for the efficient use of increased data.

Keywords: Semi-supervised learning, deep learning, natural language processing,

optimization, self-training, co-training, tri-training, tri-training with disagreement, sequencing of examples

ÖZET

Yarı Gözetimli Öğrenmede Örneklerin Sıralanması

Talha Bacak

Bilgisayar Mühendisliği Bölümü
Bitirme Projesi

Danışman: Doç. Dr. Mehmet Fatih Amasyalı

Günümüzde her geçen gün veri sayısı hızlıca artmaktadır. Derin öğrenme modelleri için bu veriler oldukça önemlidir. Gözetimli öğrenmede verilerin etiketlenmesi gerekmektedir ama veriler o kadar hızlı artmakta ki bu veriler yeterince etiketlenemiyor. Bunun için yarı gözetimli algoritmaların önemi artmaktadır. Bu algoritmaların hangi durumlarda ne kadar başarılı olduğu da önem kazanmaya başlıyor. Bu çalışmada self-training, co-training, tri-training ve tri-training with disagreement algoritmalarını inceledik. Bu algoritmaların etiketsiz veri boyutu ve eğitim verisinin boyutunun etkileri de incelenmiştir. Bu algoritmalar için bir kütüphane yazılmıştır. Bu kütüphane ile testler yapılmıştır. 5 ayrı veri seti ile hem metin sınıflandırması hem de görüntü sınıflandırması için çalışmalar yürütülmüştür. Yapılan çalışmalar sonucunda eğitim verisinin boyutu arttıkça genel olarak başarım artışı gözlemlenmiştir. Tri-training ve tri-training with disagreement algoritmalarının etiketsiz veri boyutundan çok fazla etkilendiği görülmüştür ve bu 2 algoritmanın eğitim süreleri uzun zamanlar almıştır. Tri-training algoritması başarım artışı genel olarak sağlayamamıştır. Tri-training with disagreement algoritması uygun şartlarda başarım artışı sağlamıştır. Self-training ve co-training alforitmaları hiçbir durumda başarımı olumsuz etkilememesine rağmen her durumda başarı artışı sağlayamamıştır. Veri setlerine göre başarım artışı sağlama durumu çok değişiklik göstermiştir. Bu algoritmalar ve diğer yarı gözetimli algoritmaların etkilendiği durumlar tespit edildikçe en uygun şartlarda en uygun algoritmalar optimize edilebilecek bir duruma gelecektir. Artan verilerin verimli bir şekilde kullanılması için önemli olacaktır.

Anahtar Kelimeler: Yarı gözetimli öğrenme, derin öğrenme, doğal dil işleme,

optimizasyon, self-training, co-training, tri-training, tri-training with disagreement, örneklerin sıralanması

1
Introduction

The use of machine learning algorithms, which have become widespread today, has become widespread in the classification of images and texts. These algorithms are mostly used supervised or unsupervised.

As the data we have grows every day, the cost of tagging this data increases. For this, semi-supervised algorithms are used. Our need for these algorithms continues to increase day by day.

In this project, it is aimed to write a library in which semi-supervised algorithms can be applied. By using this library, it is foreseen to determine the algorithms and the sorting situations in which these algorithms are most effective.

Self-training, co-training, tri-training and tri-training with disagreement algorithms from semi-supervised algorithms will be examined. The performance of these algorithms in different situations will be tested. For this, the effect of the training dataset will be determined by changing the dimensions of the training dataset. By changing the size of unlabeled datasets, the effect of unlabeled dataset will be seen. The running times of the algorithms will be determined. T test will be used to determine the performance.

The working situations where the algorithms are the most optimal will be determined. It will be a study that will guide the effective use of these algorithms. With such studies, it is aimed to increase the performance by testing the algorithms in different situations. A library will be written for testing, easy use and faster development of these algorithms.

The experimental results in this study will be included in the Experimental Results chapter.

2
Literature Review

In recent years, the success of deep learning and machine learning projects has increased. The most important reason for this may be that data collection has become easier with the widespread use of social media today. The cost of tagging the large amount of data generated has increased significantly. For this, semi-supervised algorithms that work with some labeled and some unlabeled data are emphasized. Some of the prominent studies in this context are summarized below.

In the first article, tri-training algorithms, one of the semi-supervised methods, are emphasized. Self-training, tri-training with disagreement, asymmetric tri-training, multi-task tri-training algorithms were compared. Domain shift methods have also been tried in this regard. [1]

When we examined another article, it was observed that Π-model and temporal ensembling algorithms from self-ensembling methods were discussed and these methods reduced the error rate considerably. [2]

It is known that augmentation methods are important for increasing the performance of semi-supervised methods. For this, Qizhe Xie et al. used switchout and back-translation methods for texts in their study and found that these reduced the error rates.[3]

In many articles, semi-supervised methods have been tried for images, and it is striking that not much work has been done on this subject for texts.

3
Feasibility

The aim of the feasibility is to examine the possibility of the project. For this purpose, in this section, the economic, legal, technical and time feasibility of the project is formed separately and examined.

3.1 Technical Feasibility

The software and hardware features to be used for the project were examined under this title.

3.1.1 Software Feasibility

The software requirements for the project were determined.

- Application Development Language: Python3
- Application Development Environment: Anaconda Spyder, Jupyter Notebook
- Framework and Python Libraries to Use: tensorflow, keras, pandas, numpy, matplotlib, time, re, nltk

3.1.2 Hardware Feasibility

The hardware features that we will use for our project are like the table 3.1.

Table 3.1 Hardware Feasibility

CPU:	**Intel i7-7700HQ CPU 2.80GHz**
System type:	**64-bit operating system, x64-based processor**
RAM Capacity:	**16 GB**
GPU:	**Nvidia GTX 1050**
Disk Capacity:	**1 TB HDD**

3.2 Time Feasibility

At this stage, the important dates and timeline of the project were determined. A Gantt chart showing the important dates and the extent to which the planned progress has been achieved is included in Figure 3.1.

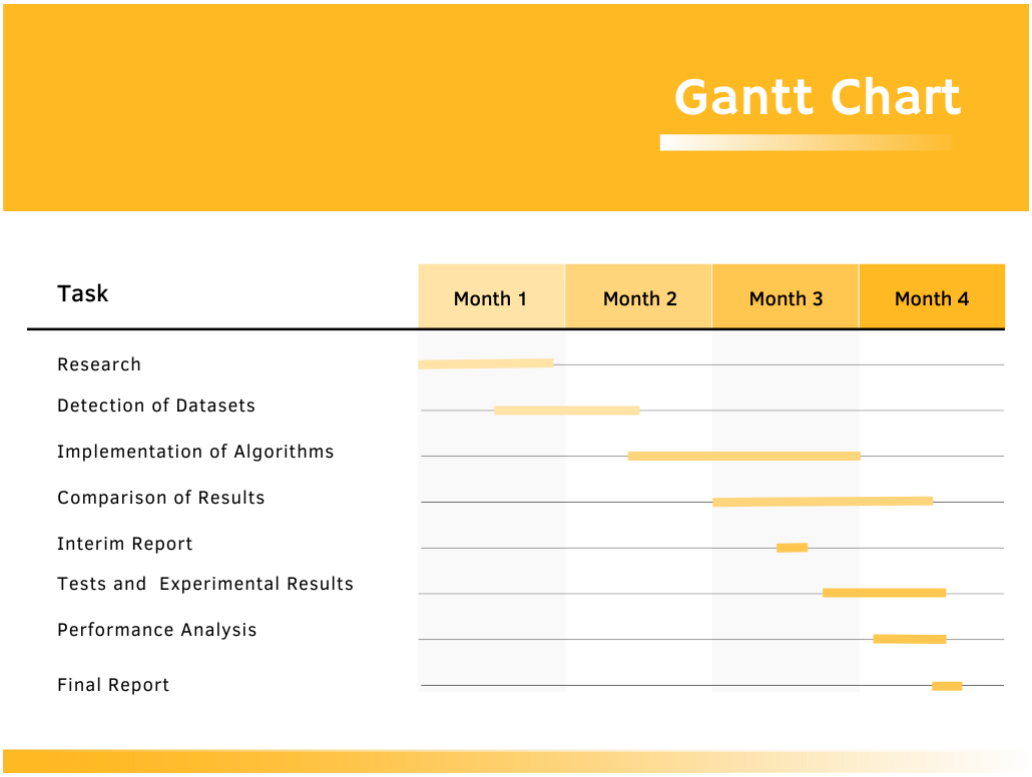

Figure 3.1 Gantt Chart

The organization chart of the activities, which are also the subject of the project report, is as in Figure 3.2.

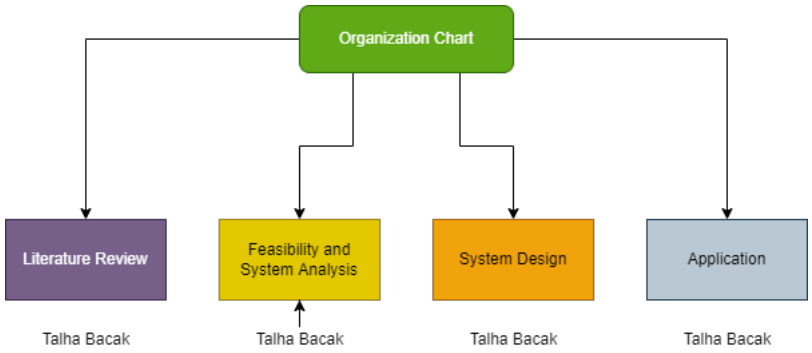

Figure 3.2 Organization Chart

3.3 Economic Feasibility

Since all software, libraries and datasets used in the project are open source, free versions and the hardware devices used during the development of the project are personal devices, there is no financial obligation in this project as the project developers do not demand any fees.

3.4 Legal Feasibility

Most of the programs used in the project are open source codes. Personal dataset was not used in this project. Also this project is supported by Yildiz Technical University.

4
System Analysis

In this project, a software library will be created. This library is intended to support tensorflow keras models. These models will be made workable with some semi-supervised algorithms. The results obtained with this library will be compared and appropriate situations will be determined.

4.1 The goal of the project

In this project, some semi-supervised algorithms will be examined and a software library will be created. The performances of these algorithms for text classification will be determined. It will be determined in which order these algorithms give more appropriate results.

4.2 Draft Diagram

It is used to represent the situation between the system and the entities in relation to it. Here the system is defined as "Library", the entity is defined as "Dataset", "Model" and "Result". The created design is as in Figure 4.1

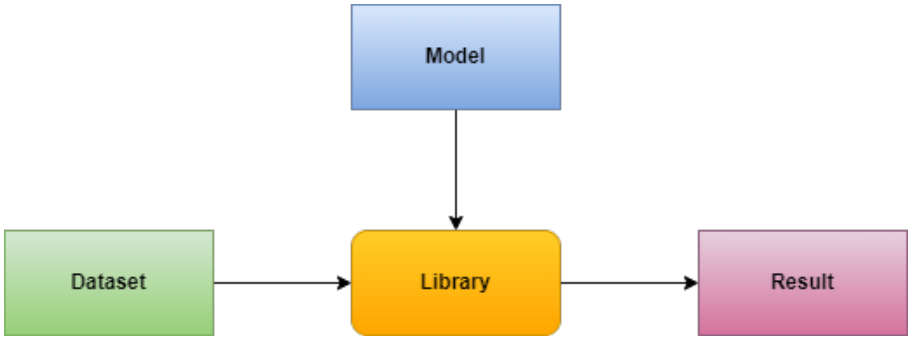

Figure 4.1 Draft Diagram

4.3 Data Flow Diagram

In this section, the relationship between the entities determined in the draft diagram and the system is visualized in the figure 4.2 by further detailing according to the processes that will develop.

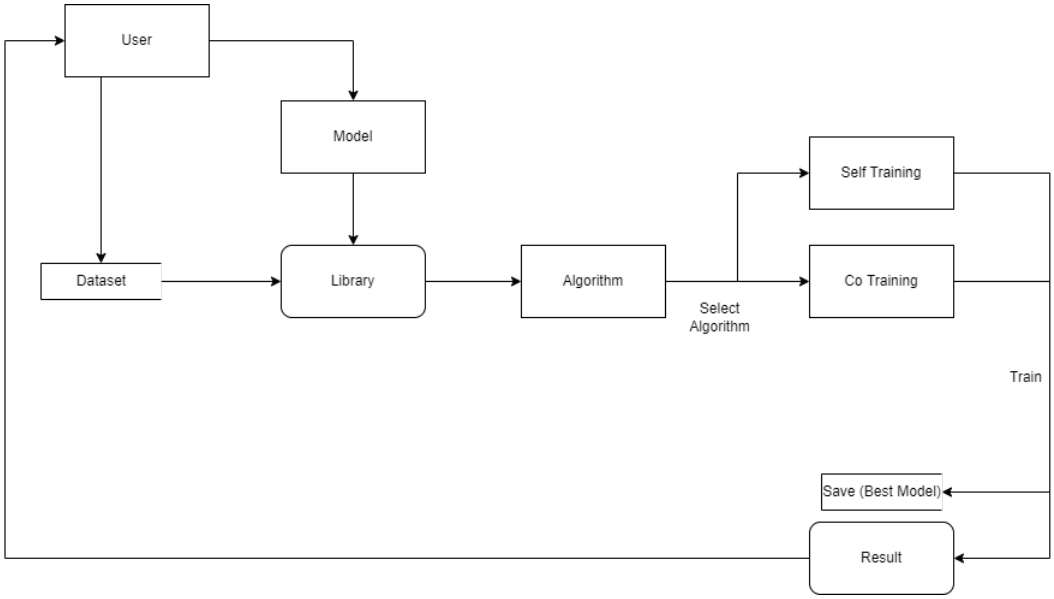

Figure 4.2 Data Flow Diagram

4.4 Use Case Diagram

The basic information about the operation of the design is schematized in the figure 4.3 by creating a use case diagram that explains the operation of the project and the event.

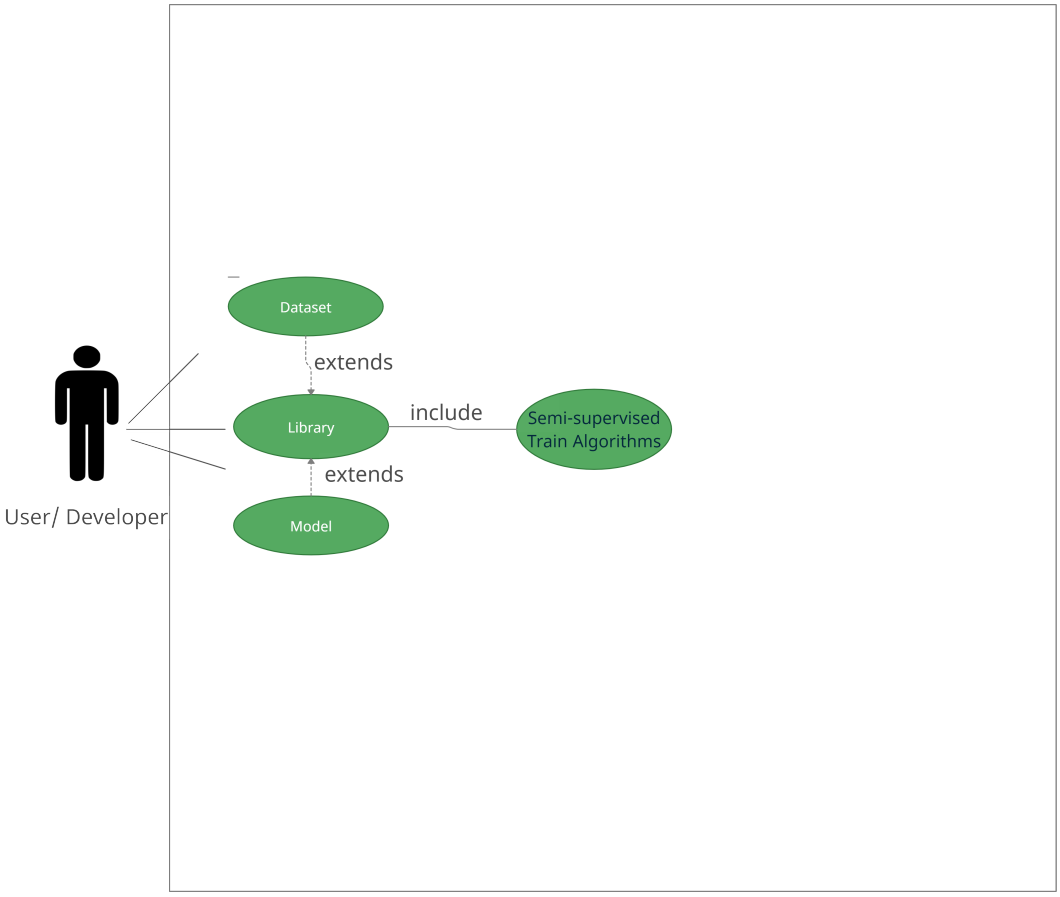

Figure 4.3 Use-Case Diagram

5
System Design

5.1 Software Design

In the software design, the main operations of the system are determined, and the figure is shown in the state diagram as in 6.3. Later, these main processes are explained in detail.

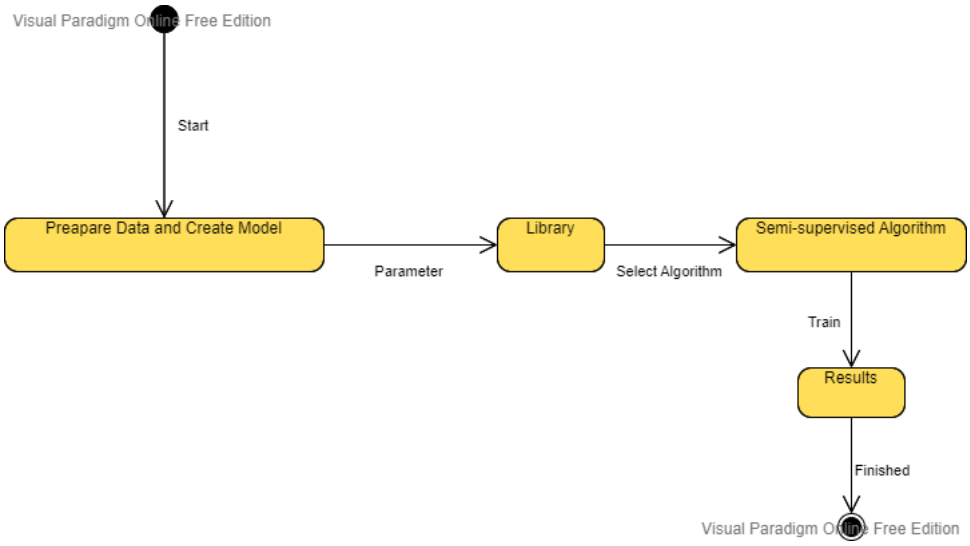

Figure 5.1 State diyagramı

5.1.1 Prepare Data and Create Model

Data can be images or text. For these 2, 2 different data preparation classes have been added to the library. Data should be prepared with these classes. After that, the user should create a keras model. The figure is shown as in 6.3.

```
data2 = DataSet(train2, valid2, test2,unlabel=unlabel,haveUnlabel_y=True)
data2.setData()
data1 = DataSet(train1, valid1, test1, unlabel=unlabel,haveUnlabel_y=True)
data1.setData()

def create_model1(size_of_vocabulary):
    model = Sequential()
    model.add(Embedding(size_of_vocabulary,128,input_length=120))
    model.add(Bidirectional(LSTM(16,return_sequences=True,dropout=0.2)))
    model.add(GlobalMaxPooling1D())
    model.add(Dense(16,activation='relu'))
    model.add(Dropout(0.05))
    model.add(Dense(1,activation='sigmoid'))
    model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy', Precision(), Recall()])
    model.summary()
    return model

def create_model2(size_of_vocabulary):
    model = Sequential()
    model.add(Embedding(size_of_vocabulary,128,input_length=120))
    model.add(SpatialDropout1D(0.4))
    model.add(LSTM(32, dropout=0.2, recurrent_dropout=0.2))
    model.add(Dense(1,activation='sigmoid'))
    model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy', Precision(), Recall()])
    model.summary()
    return model

model1 = create_model1(data1.size_of_vocabulary)
model2 = create_model2(data2.size_of_vocabulary)
```

Figure 5.2 For example

5.1.2 Semi-supervised Algorithm

Some semi-supervised algorithms are available in the library. These are algorithms such as Self Training, Co Training. Models can be trained with these algorithms.

5.1.3 Result

As a result of the training, the performance of the models can be determined. We can compare these results with the performances of the situations and algorithms we have determined.

5.2 Database Design

The developer/user can use it with text and photo data that the developer/user will create or use ready-made.

5.3 Input-Output Design

Giving dataset and model as input are mandatory parameters. As output, model file and performance results are given.

6
Application

The library written within the scope of the project includes the following algorithms:

- Self-training
- Co-training
- Tri-training
- Tri-training with disagreement

Apart from these algorithms, a library was also written for the data set:

- For preprocessed, ready-made datasets,
- For unprocessed text datasets,

two different classes were writed.

The library supports tensorflow and keras. It supports sigmoid and softmax as output activation function. The user can specify the activation function of the model and how many epochs to run.

6.1 Self-training

In this algorithm, unlabeled data are labeled according to the most successful predictions at each iteration. For example, let's train 10 epochs in a situation where there are 2000 unlabeled data. In this case, the top 200 unlabeled data are labeled for each epoch result. Success is determined by the highest prediction rate of the activation function. The tagged data is added to the training dataset. One model is used.

Algorithm 1 Self-training

repeat
  $m \leftarrow train\_model(L)$
  for $x \in U$ **do**
    if $\max m(x) > \tau$ **then**
      $L \leftarrow L \cup \{(x, p(x))\}$
until no more predictions are confident

Figure 6.1 General self-training algorithm [4]

6.2 Co-training

In this algorithm, as in self-training, unlabeled data are labeled according to the most successful predictions at each iteration. The difference of this algorithm is that it works with 2 models instead of one model. Unlabeled data, where the first model under-estimates but the second model over-estimates, is added to the training dataset of the first model. Unlabeled data, where model 2 underestimates but model 1 overestimates, is added to the training dataset of model 2.

Algorithm 2 Co-training

repeat
  $m_1 \leftarrow train\_classifier(L^1)$
  $m_2 \leftarrow train\_classifier(L^2)$
  for $x \in U$ **do**
    if $\max m_1(x) > \tau$ **and** $\max m_2(x) < \tau$ **then**
      $L^2 \leftarrow L^2 \cup \{(x, p_1(x))\}$
    if $\max m_2(x) > \tau$ **and** $\max m_1(x) < \tau$ **then**
      $L^1 \leftarrow L^1 \cup \{(x, p_2(x))\}$
until no more predictions are confident based on *one* classifier

Figure 6.2 General co-training algorithm [5]

6.3 Tri-training

Three models are used in this algorithm. It is slightly different from self-training and co-trining. The 3 models are first trained with bootstrap samples. After this training, the following steps are performed in each iteration:

- All unlabeled data are estimated with the other 2 models.
- If the predicted classes are the same, they are labeled and added to a cluster.
- With this set of labeled data and original labelled data, the model is trained.

- These operations are done for all 3 models.
- All of the above operations are repeated for the specified number of epochs.

Algorithm 3 Tri-training

for $i \in \{1..3\}$ **do**
  $S_i \leftarrow bootstrap\_sample(L)$
  $m_i \leftarrow train\_model(S_i)$
repeat
  for $i \in \{1..3\}$ **do**
    $L_i \leftarrow \emptyset$
    for $x \in U$ **do**
      if $p_j(x) = p_k(x)(j, k \neq i)$ **then**
        $L_i \leftarrow L_i \cup \{(x, p_j(x))\}$
    $m_i \leftarrow train\_model(L \cup L_i)$
until none of m_i changes
apply majority vote over m_i

Figure 6.3 General tri-training algorithm [6]

6.4 Tri-training with disagreement

This algorithm is quite similar to the tri-training algorithm. Unlike tri-traing, when deciding on the label of unlabeled data, in addition to the estimation of the other 2 models, it also looks at the model's difference from its own estimation.

7
Experimental Result

In this study, studies were carried out with both text and image datasets.

7.1 Experimental Data Exploration

In this study, IMDB review, financial and tweets data sets were used as text. Mnist and fashion mnist data sets were used as images. You can have a look at 7.1 about datasets. The datasets indicated with "(s)" in the tables represent datasets with scaled down training set.

Table 7.1 Dataset information

	IMDB	**financial**	**tweets**	**mnist**	**fashion mnist**
Train:	991	422	4903	5000	5000
Small Train:	496	211	2452	1000	1000
Test:	989	88	981	1000	1000
Unlabel:	3963	963	14687	24000	24000
Class:	2	2	2	10	46

7.2 Supervised Train (without Unlabel Data)

These data sets were tested with the algorithms in the developed library. Without using these algorithms, supervised training results with labeled data are as in 7.2 and 7.3. There are 20 epochs trained results in 7.2. There are 200 epochs trained results in 7.3.

The reason for training 20 epochs and 200 epochs: It was made to compare the situations in the same time period due to the longer duration of the semi-supervised algorithms.

Table 7.2 20 epochs train

	F1 Score (mean+-standard deviation)	Time (mean)(second)
IMDB:	75,6+-1,43	67
financial:	83,74+-0,44	25
tweets:	82,29+-0,71	352
mnist:	95,47+-0,55	32
fashion mnist:	85,42+-0,75	36
(s) IMDB:	71,63+-3,74	32
(s) financial:	83,74+-1,75	11
(s) tweets:	80,88+-0,49	113
(s) mnist:	86,25+-2,36	11
(s) fashion mnist:	81,88+-0,75	13

Table 7.3 200 epochs train

	F1 Score (mean+-standard deviation)	Time (mean)(second)
IMDB:	72,16+-3,25	1662
financial:	83,14+-1,33	362
tweets:	80,96+-0,68	3442
mnist:	95,2+-0,61	299
fashion mnist:	84,26+-1,59	281
(s) IMDB:	72,66+-2,56	477
(s) financial:	83,07+-0,5	135
(s) tweets:	79,72+-0,26	3558
(s) mnist:	86,07+-2,45	128
(s) fashion mnist:	82,07+-1,15	124

7.3 Semi-supervised Train (with Unlabel Data)

In semi-supervised algorithms, trainings were also made by changing the proportions of unlabeled data. The table is proportioned according to the number of unlabels given in 7.1. To better understand the tables in the following sections, read the information below.

- 1/4 : Unlabel is trained with a quarter of the data.
- 2/4 : Unlabel is trained with half of the data.
- 3/4 : Unlabel is trained with 3/4 of the data.
- 4/4 : Unlabel is trained with all of the data.
- Time data is the time data of 4/4. (mean)(second)
- Results are F1 Scores. (mean+-standard deviation)

- Green color: Significant increase in success compared to the t-test.
- Red color : Significant decrease in success compared to the t-test.

7.3.1 Self-Training

Self-training test results are as in the table 7.4:

Table 7.4 Self-training results (20 epochs)

	1/4	2/4	3/4	4/4	Time
IMDB:	75,24+-2,19	76,12+-1,24	76,77+-1,29	76,95+-1,65	236
financial:	84,7+-0,59	85,3+-0,76	85,1+-0,77	84,99+-0,61	82
tweets:	82,13+-0,78	81,76+-0,91	82,68+-0,43	82,81+-1,03	1553
mnist:	95,35+-0,31	95,00+-1,35	95,37+-0,66	95,32+-0,52	557
fashion mnist:	86,63+-0,63	86,56+-0,58	86,86+-0,44	86,5+-0,53	463
(s) IMDB:	72,31+-3,52	73,45+-0,92	74,43+-0,97	75,07+-1,1	119
(s) financial:	84,46+-1,03	82,47+-1,44	84,39+-1,35	83,61+-1,51	28
(s) tweets:	82,19+-0,77	80,99+-0,71	81,23+-0,8	80,72+-1,01	549
(s) mnist:	87,51+-0,62	87,22+-0,6	86,21+-1,37	87,08+-1,25	381
(s) fashion mnist:	81,82+-1,04	81,01+-0,95	81,78+-0,71	81,53+-0,9	499

7.3.2 Co-Training

The test results related to co-training are as in the table 7.5:

Table 7.5 Co-training results (20 epochs)

	1/4	2/4	3/4	4/4	Time
IMDB:	76,87+-1,21	75,87+-2,28	76,83+-1,89	76,05+-2,3	443
financial:	84,94+-1,07	85,84+-1,39	84,51+-1,86	85,4+-1,88	118
tweets:	82,93+-0,5	82,96+-0,52	82,39+-0,69	82,59+-0,79	2333
mnist:	95,54+-0,26	95,69+-0,33	95,69+-0,32	95,8+-0,37	648
fashion mnist:	86,1+-0,59	84,89+-1,7	85,18+-1,41	85,11+-0,97	561
(s) IMDB:	73,51+-2,02	74,09+-2,41	74,97+-1,87	74,07+-1,85	203
(s) financial:	83,66+-0,83	82,89+-0,56	83,72+-1,47	83,04+-1,56	71
(s) tweets:	81,53+-0,37	82,01+-0,96	80,87+-1,16	81,63+-0,47	1122
(s) mnist:	87,63+-1,99	86,73+-1,89	86,79+-2,56	88,24+-0,49	488
(s) fashion mnist:	81,09+-2,14	81,57+-1,06	80,14+-1,39	80,43+-1,64	486

7.3.3 Tri-Training

Tri-training test results are as in the table 7.6:

In the IMDB dataset, there is overfitting at the bootstrap stage. So ignore the IMDB results in the tri-training table 7.6. .

Table 7.6 Tri-training results (20 epochs)

	1/4	2/4	3/4	4/4	Time
IMDB:	52,99+-8,61	21,57+-3,71	23,25+-15,65	22,37+-17,63	1008
financial:	86,98+-0,98	84,62+-1,4	82,74+-0,31	81,48+-0,46	673
tweets:	82,34+-0,49	82,69+-0,69	82,97+-0,43	82,57+-0,73	10320
mnist:	95,37+-0,31	95,83+-0,29	95,84+-0,45	95,35+-0,43	1287
fashion mnist:	85,47+-0,63	86,08+-0,76	86,12+-0,47	85,84+-1,25	1167
(s) IMDB:	30,37+-21,38	23,97+-23,95	19,81+-30,05	25,87+-21,43	1648
(s) financial:	83,9+-0,37	84,05+-0,52	83,61+-1,07	80,91+-0,36	480
(s) tweets:	80,04+-0,25	80,94+-0,27	80,5+-0,18	80,67+-0,23	6841
(s) mnist:	87,57+-2,55	86,26+-3,63	84,29+-3,3	84,16+-1,36	912
(s) fashion mnist:	74,63+-4,69	75,58+-3,56	72,58+-2,56	70,48+-2,87	860

7.3.4 Tri-Training with Disagreement

The test results regarding tri-training with disagreement are as in the table 7.7:

Table 7.7 Tri-training with disagreement results (20 epochs)

	1/4	2/4	3/4	4/4	Time
IMDB:	76,67+-1,59	76,07+-1,47	77,65+-2,26	75,93+-2,61	988
financial:	84,74+-1,05	84,97+-1,1	85,51+-1,38	85,31+-0,47	349
tweets:	83,39+-0,32	83,44+-0,52	83,96+-0,29	83,11+-0,58	3328
mnist:	95,69+-0,42	95,88+-0,35	95,95+-0,84	95,79+-0,59	1287
fashion mnist:	85,42+-1,05	86,31+-0,78	85,95+-1,09	86,28+-0,5	833
(s) IMDB:	74,55+-2,44	70,59+-4,33	61,3+-17,77	65,61+-3,13	831
(s) financial:	83,28+-1,48	84,29+-1,39	83,97+-0,8	80,65+-4,04	242
(s) tweets:	81,62+-0,79	82,19+-0,64	82,01+-0,6	82,01+-0,41	3193
(s) mnist:	89,21+-1,74	88,61+-1,46	87,43+-1,22	84,63+-1,75	676
(s) fashion mnist:	81,32+-0,74	81,37+-0,69	80,89+-2,61	79,11+-2,06	884

7.4 Behavior of tagging unlabeled data

An extra study was performed with the Turkish data set. In this study, labeling status of unlabeled data was examined.

7.4.1 Data Exploration

The information of this dataset is as in the table 7.8 and figure 7.1. The class distribution of the data is in the figure 7.1.

Table 7.8 Dataset information

	Turkish Sentiment Analysis Dataset
Train:	991
Small Train:	496
Test:	989
Unlabel:	3963
Class:	2

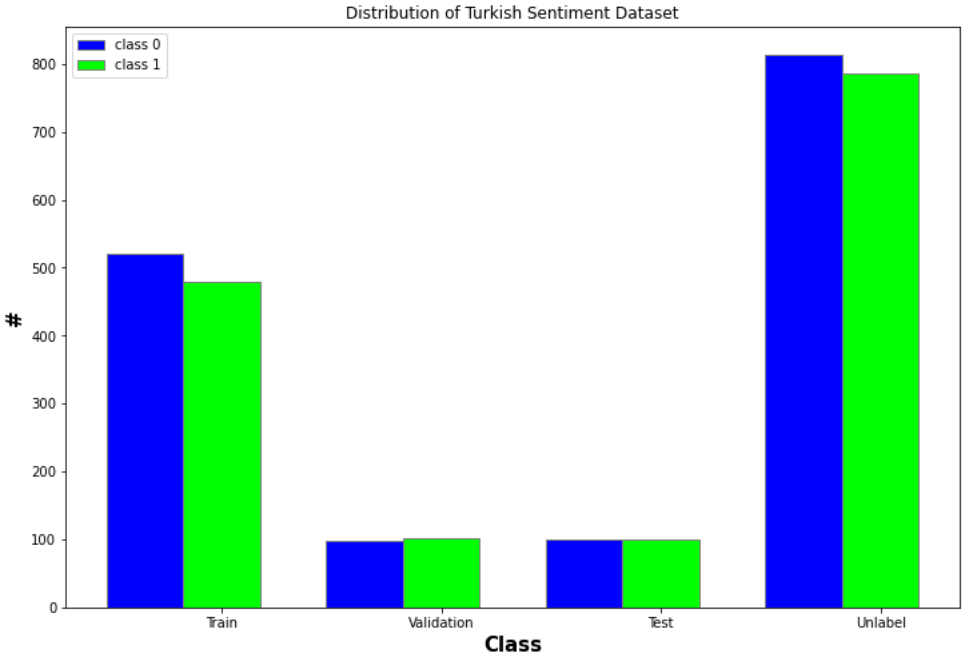

Figure 7.1 Distribution of Dataset

7.4.2 Examining unlabeled data

The information of self-training, co-training, tri-training and tri-training with disagreement algorithms trained with Turkish dataset during training is figure 7.2, figure 7.3, figure 7.4 and figure 7.5 as in the pictures. The graphs in these pictures contain the information of the unlabeled data. Green indicates unlabeled data is correctly labeled, red indicates incorrectly labeled data. Light colored ones belong to 0th class, dark colored ones belong to 1st class. It is the data added to the training set by labeling from unlabeled data in each epoch.

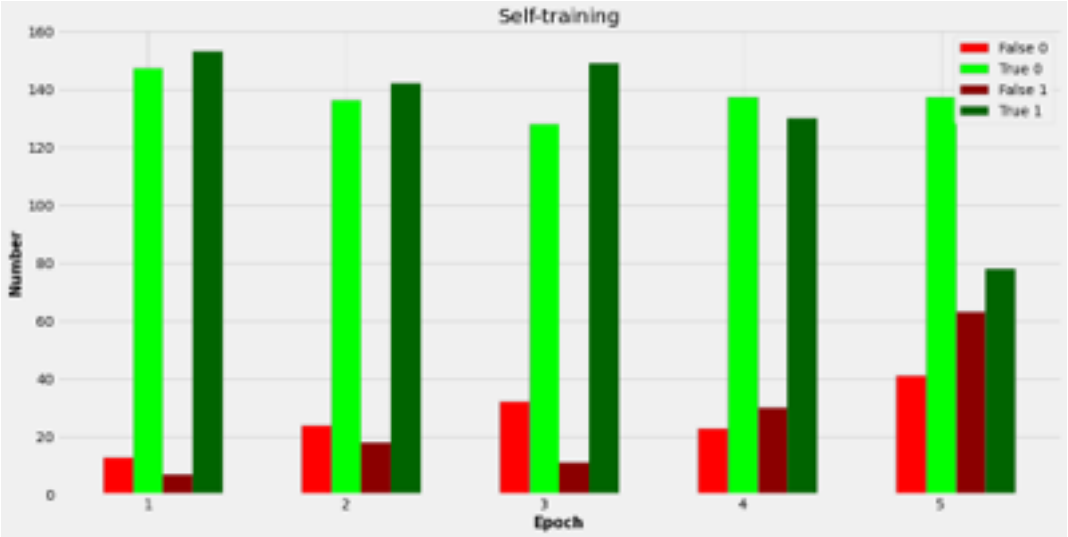

Figure 7.2 Self-training

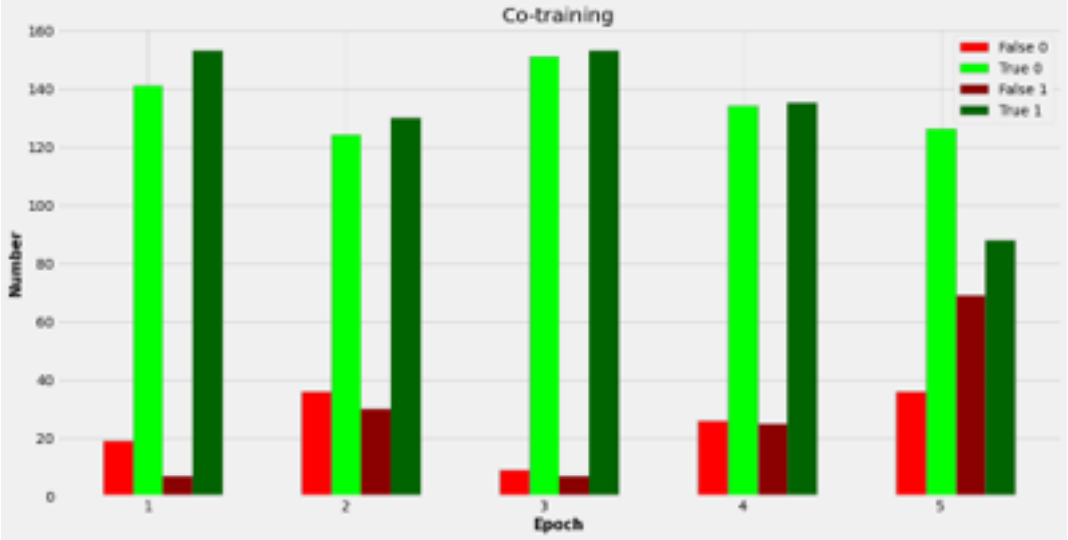

Figure 7.3 Co-training

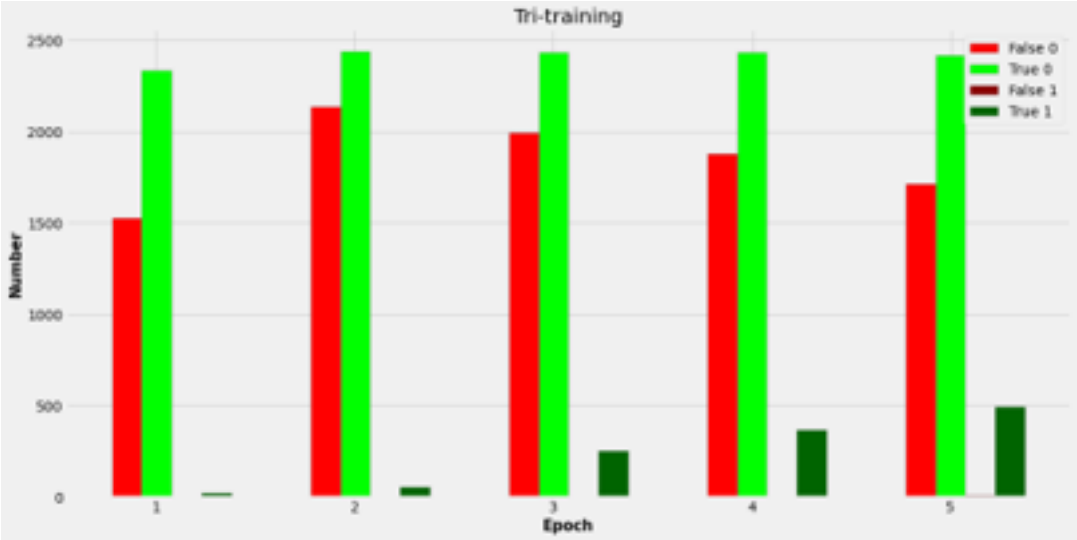

Figure 7.4 Tri-training

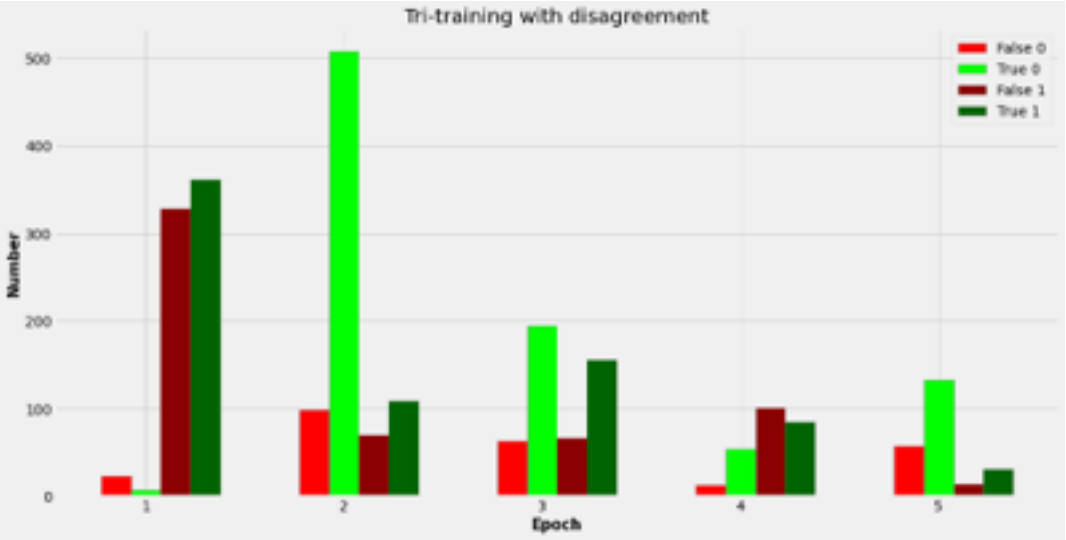

Figure 7.5 Tri-training with disagreement

8
Performance Analysis

The experimental results of the algorithms in the created library will be examined in this section.

8.1 Comparison

8.1.1 Time

Let's examine it temporally:

- Tri-training was the longest running algorithm.
- tri-training with disagreement took slightly less time than tri-training.
- Co-training ran at least 2x faster than tri-training with disagreement.
- Self-training ran about 2x faster than co-training.

8.1.2 Memory

Let's take a look at the memory:

- Tri-training and tri-training with disagreement are the algorithms that use the most memory.

8.1.3 Training Dataset Size

When we examine it according to the size of the training dataset:

- It has been observed that there is a decrease in performance increase in general when the data set is smaller.

- In some data sets, this decrease was more, while in others it was less.
- Tri-training and tri-training with disagreement were the algorithms most affected by the size of the dataset.

8.1.4 Unlabel Dataset Size

When we examine according to the ratio of the unlabeled data set:

- The performance of tri-training and tri-training with disagreement algorithms decreased as the unlabeled data size increased too much. It has been more successful with less unlabeled data.
- Self-training and co-training were less affected by changing the size of unlabeled data. Too much or too little unlabeled data has partially affected it negatively.

8.1.5 Text or Image

Text data gave slightly better results than image data.

8.1.6 Algorithms

- Tri-training failed.
- Tri-training with disagreement can be quite successful with a small number of unlabeled data.
- Co-training and self-training increased performance on some datasets.

8.2 Behavior of Tri-training with disagreement

As seen in 8.1, 8.2 and 8.4 in this algorithm, there are many ups and downs during training. I think that it will provide a very high performance increase in long-term training. More research can be done on this subject.

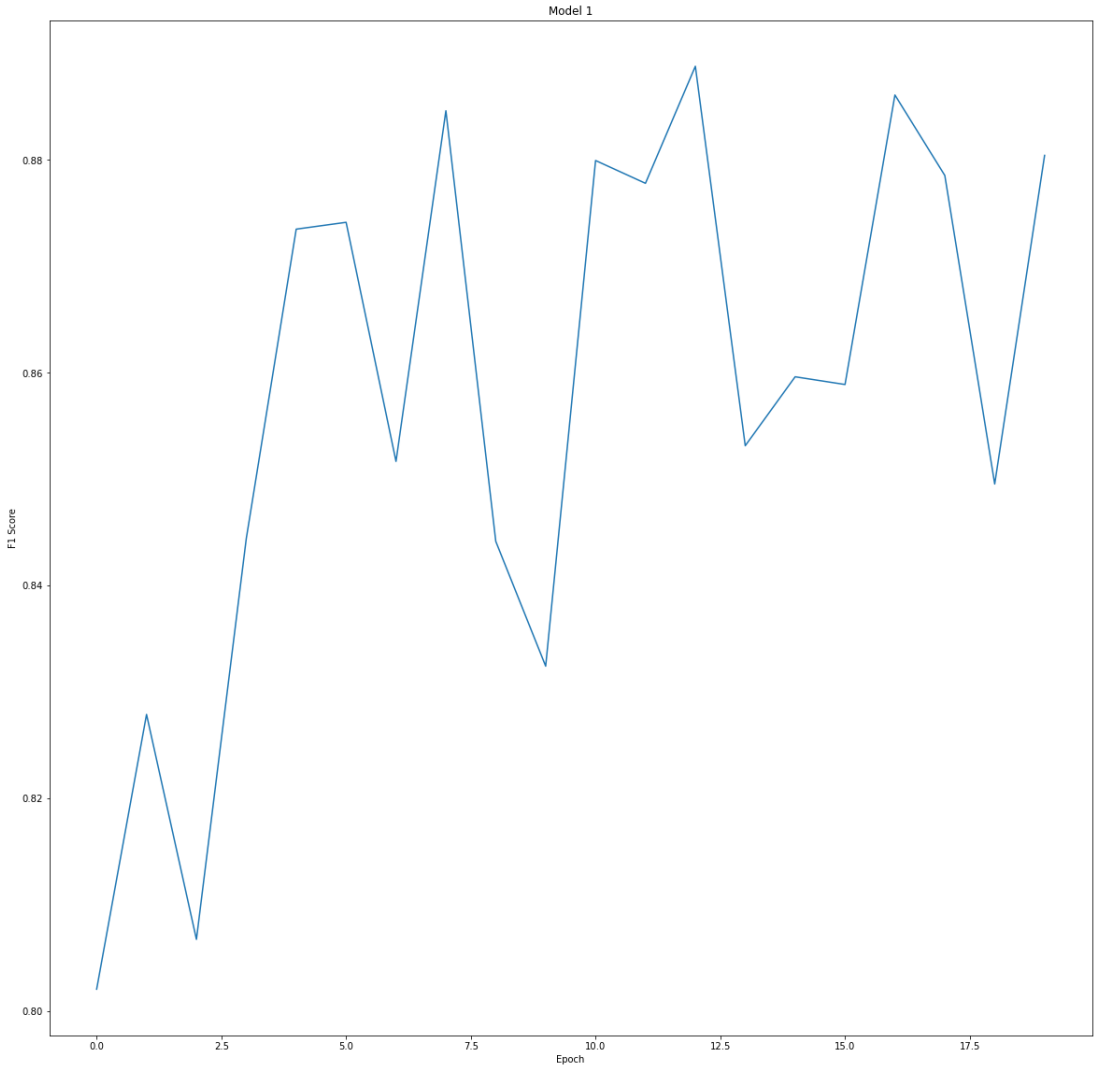

Figure 8.1 Tri-training with disagreement example 1 (small mnist)

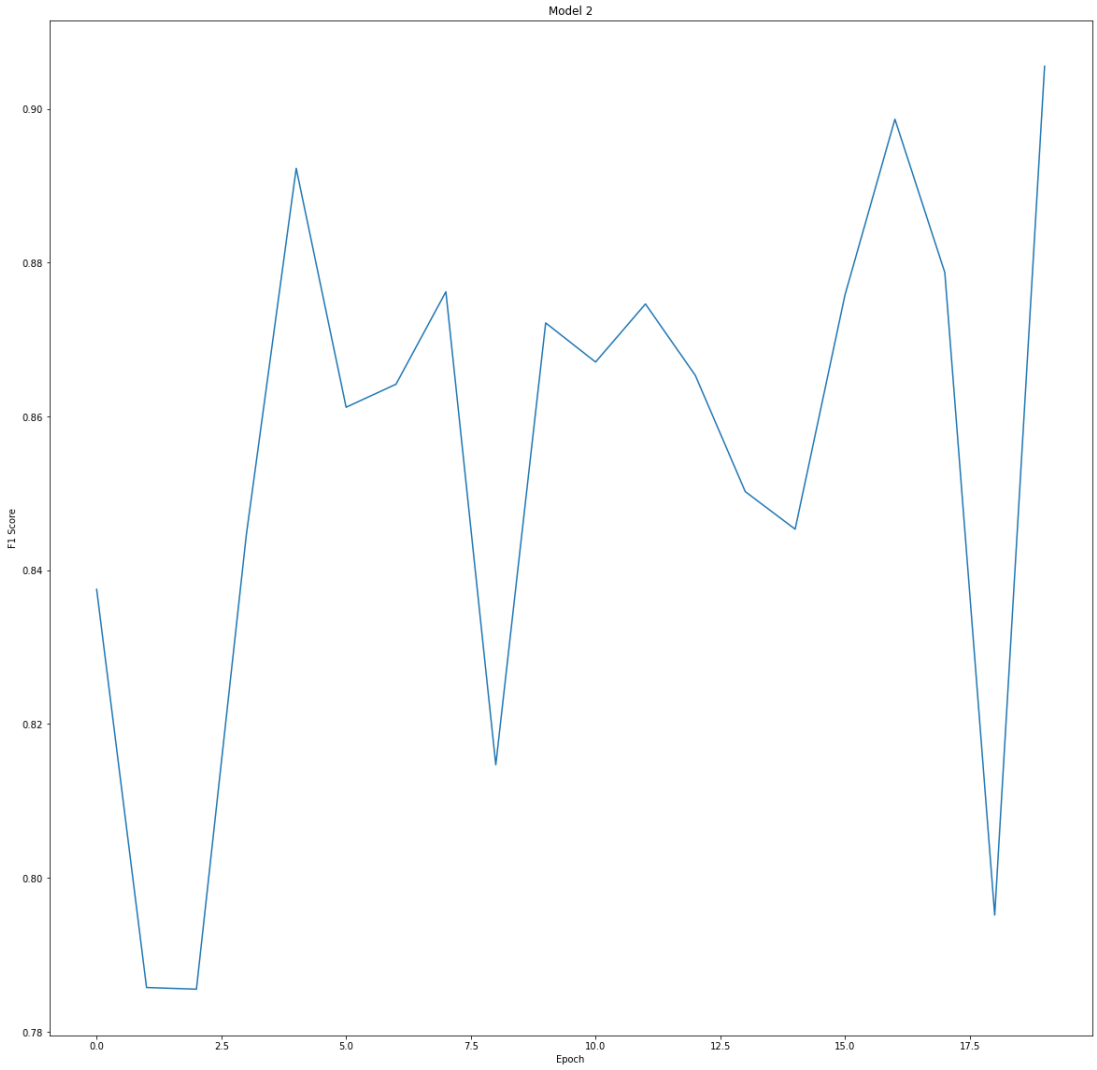

Figure 8.2 Tri-training with disagreement example 2 (small mnist)

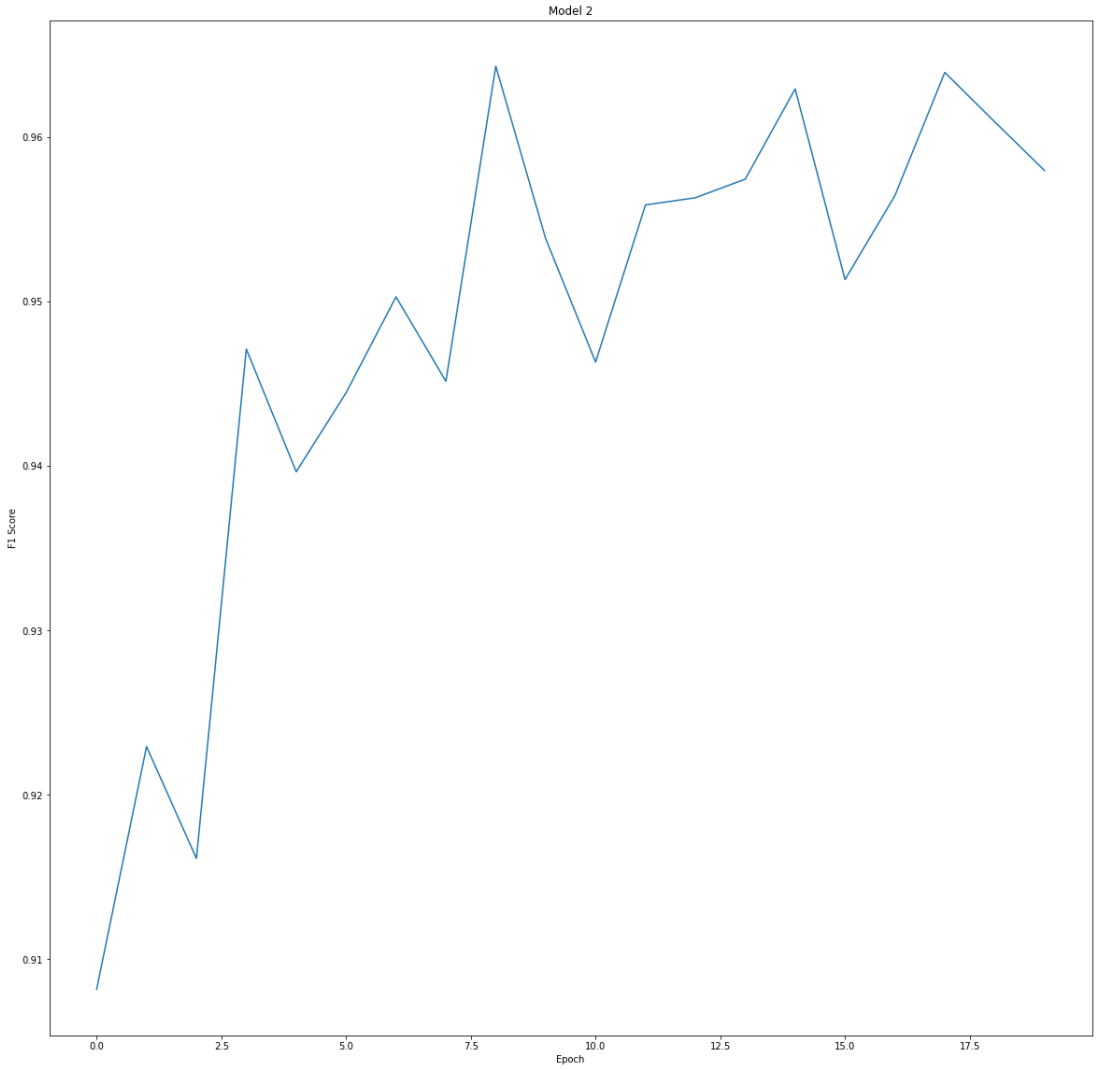

Figure 8.3 Tri-training with disagreement example 3 (mnist)

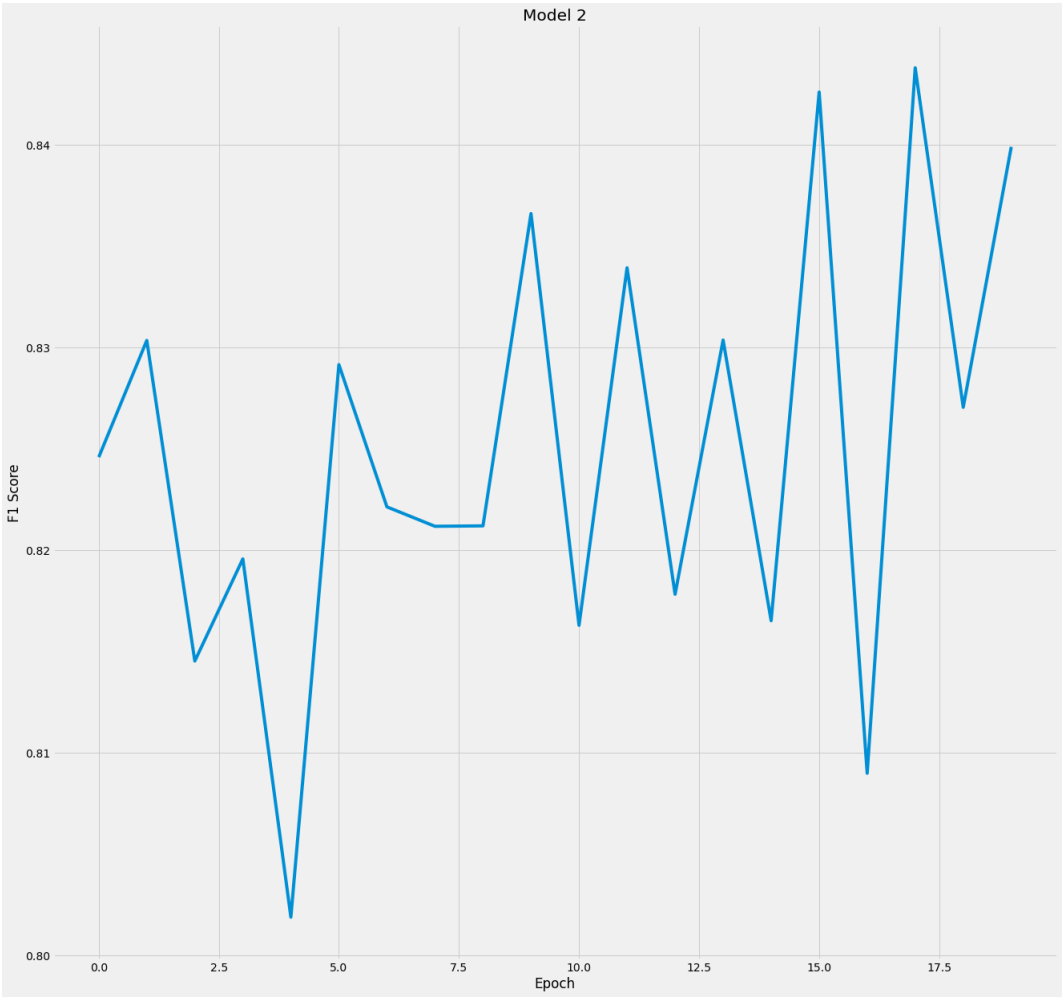

Figure 8.4 Tri-training with disagreement example 4 (tweets)

9
Result

According to the studies and experimental results, it can be said that self-training, co-training and tri-training with disagreement algorithms increase success. Tri-training algorithm could not increase performance, maybe it can increase performance with very little unlabeled data.

Tri-training with disagreement algorithm can provide high performance with unlabeled datasets that are not much larger than the training dataset.

Tri-training and tri-training with disagreement algorithms are heavily influenced by the size of unlabeled data.

Self-training and co-training algorithms did not adversely affect success in any dataset. Success will not decrease with the use of these algorithms. But in all cases, they cannot provide an increase in success.

All algorithms were generally more successful when the training data set was larger. The performance of the co-training and tri-training with disagreement algorithm was better on some small training datasets.

Tri-training and tri-training with disagreement algorithms take a long time. They also need a lot of memory.

References

[1] S. Ruder and B. Plank, "Strong baselines for neural semi-supervised learning under domain shift," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia: Association for Computational Linguistics, 2018, pp. 1044–1054. [Online]. Available: http://aclweb.org/anthology/P18-1096.

[2] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," *CoRR*, vol. abs/1610.02242, 2016. arXiv: 1610.02242. [Online]. Available: http://arxiv.org/abs/1610.02242.

[3] Q. Xie, Z. Dai, E. H. Hovy, M. Luong, and Q. V. Le, "Unsupervised data augmentation," *CoRR*, vol. abs/1904.12848, 2019. arXiv: 1904.12848. [Online]. Available: http://arxiv.org/abs/1904.12848.

[4] S. Abney, in *Semisupervised learning for computational linguistics*, 2007.

[5] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, ser. COLT' 98, Madison, Wisconsin, USA: Association for Computing Machinery, 1998, pp. 92–100, ISBN: 1581130570. DOI: 10.1145/279943.279962. [Online]. Available: https://doi.org/10.1145/279943.279962.

[6] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, pp. 1529–1541, Dec. 2005. DOI: 10.1109/TKDE.2005.186.

Curriculum Vitae

FIRST MEMBER

Name-Surname: Talha Bacak
Birthdate and Place of Birth: [redacted]
E-mail: l1116038@std.yildiz.edu.tr
Phone: +[redacted]
Practical Training: Turkiye Sigorta, STM

Project System Informations

System and Software: Python, tensorflow, keras
Required RAM: 4GB
Required Disk: 2GB