

**REPUBLIC OF TURKEY
AKDENİZ UNIVERSITY**



**AUTO GRADING OF ANSWERS FOR TEXT-BASED OPEN-ENDED
QUESTIONS USING NATURAL LANGUAGE PROCESSING**

Burak KESKİN

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER OF SCIENCE THESIS

JUNE 2022

ANTALYA

**REPUBLIC OF TURKEY
AKDENİZ UNIVERSITY**



**AUTO GRADING OF ANSWERS FOR TEXT-BASED OPEN-ENDED
QUESTIONS USING NATURAL LANGUAGE PROCESSING**

Burak KESKİN

INSTITUTE OF NATURAL AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER ENGINEERING

MASTER OF SCIENCE THESIS

JUNE 2022

ANTALYA

REPUBLIC OF TURKEY
AKDENİZ UNIVERSITY
INSTITUTE OF NATURAL AND APPLIED SCIENCES

**AUTO GRADING OF ANSWERS FOR TEXT-BASED OPEN-ENDED
QUESTIONS USING NATURAL LANGUAGE PROCESSING**

Burak KESKİN

DEPARTMENT OF COMPUTER ENGINEERING

MASTER OF SCIENCE THESIS

This thesis was accepted unanimously by the jury on 17/06/2022.

Prof. Dr. Melih GÜNAY (Supervisor)

Asst. Prof. Dr. Murat AK

Asst. Prof. Dr. Özge Öztimur KARADAĞ

ÖZET

DOĞAL DİL İŞLEME KULLANARAK METİN TABANLI AÇIK UÇLU SORULAR İÇİN CEVAPLARIN OTOMATİK NOTLANDIRILMASI

Burak Keskin

Yüksek Lisans Tezi, Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Prof. Dr. Melih GÜNAY

Haziran 2022; 39 sayfa

Bu tez kapsamında, Python programlama dili üzerinde geliştirilmiş çeşitli teknolojiler kullanılarak metin tabanlı açık uçlu soruların cevapları için otomatik notlandırma yapan bir Python modülü geliştirilmiştir. Modülün geliştirilme sürecinde, öğrencilerin verdiği cevaplar ile çözüm anahtarında yer alan cevabın karşılaştırılması çeşitli makine öğrenmesi ve doğal dil işleme teknikleri kullanılarak gerçekleştirilmiştir.

Modül tarafından analiz edilen cevaplar, önceden eğitilmiş modeller yardımı ile vektörel düzleme aktarıldıktan sonra cevap anahtarı ile karşılaştırılmıştır. Bu tez kapsamında Doğa Bilimleri (Natural Science) dersinin verileri bu karşılaştırma için seçilmiştir.

ANAHTAR KELİMELEER: Python, Dönüştürücüler, Doğal Dil İşleme, Bilgisayar Tabanlı Değerlendirme

JÜRİ: Prof. Dr. Melih GÜNAY

Dr. Öğr. Üyesi Murat AK

Dr. Öğr. Üyesi Özge Öztimur KARADAĞ

ABSTRACT

AUTO GRADING OF ANSWERS FOR TEXT-BASED OPEN-ENDED QUESTIONS USING NATURAL LANGUAGE PROCESSING

Burak KESKİN

MSc Thesis in COMPUTER ENGINEERING

Supervisor: Prof. Dr. Melih GÜNAY

JUNE 2022; 39 pages

Within the scope of this thesis, a Python module that automatically grades the answers to text-based open-ended questions has been developed using various technologies developed on the Python programming language. During the development of the module, the comparison of the answers given by the students with the answer in the solution key was carried out using various machine learning and natural language processing techniques.

The answers analyzed by the module were compared with the answer key after they were transferred to the vector plane with the help of pre-trained models. Within the scope of this thesis, the data of the Natural Science course was selected for this comparison.

KEYWORDS: Python, Transformers, Natural Language Processing, Computer Based Assessment

COMMITTEE: Prof. Dr. Melih GUNAY

Asst. Prof. Dr. Murat AK

Asst. Prof. Dr. Özge Öztimur KARADAĞ

ACKNOWLEDGEMENTS

First and foremost, I would like to express my very great appreciation to Prof. Dr. Melih GÜNAY for unwavering guidance and patience that cannot be underestimated during this study.

Finally, I would like to thank my family since they have supported me all of my life.



LIST OF CONTENTS

ÖZET	i
ABSTRACT	ii
ACKNOWLEDGEMENTS	iii
TEXT OF OATH	vi
ABBREVIATIONS	vii
List of Figures	viii
List of Tables	ix
1. INTRODUCTION	1
2. LITERATURE REVIEW	3
2.1. Computer Based Assessment	3
2.2. Natural Language Processing	4
2.2.1. NLTK	5
2.2.2. TF-IDF	5
2.2.3. T5	6
2.2.4. Transformers	6
2.2.5. BERT	7
2.2.6. Universal Sentence Encoder	8
2.3. Similarity Measures	14
2.3.1. Cosine Similarity	14
2.3.2. Z-Score	14
3. MATERIAL AND METHOD	15
3.1. Python Programming Language	15
3.2. Visual Studio Code	15
3.3. Pandas Library	16
3.4. Numpy Library	16
3.5. Scipy Library	16
3.6. Tensorflow	17
3.7. BERT	17
3.8. Universal Sentence Encoder	18
3.9. Git	20
3.10. Computer Specifications	22
4. RESULTS AND DISCUSSION	23

5. CONCLUSION	37
6. REFERENCES	38
ÖZGEÇMİŞ	



TEXT OF OATH

I declare that this study "Auto Grading of Answers for Text-Based Open-Ended Questions Using Natural Language Processing ", which I present as master thesis, is in accordance with the academic rules and ethical conduct. I also declare that i cited and referenced all material and results that are not original to this work.

17/06/2022

Burak KESKİN



ABBREVIATIONS

BERT	: Bidirectional Encoder Representations from Transformers
USE	: Universal Sentence Encoder
NLTK	: Natural Language Toolkit
DAN	: Deep Averaging Network
CBA	: Computer Based Assessment
T5	: Text-to-Text Transfer Transformer
SQUAD	: Stanford Question and Answering Database



List of Figures

Figure 2.1.	Diagram of the text-to-text framework (Haller 2020)	6
Figure 2.2.	Representation of Transformer (Vaswani et al. 2017)	7
Figure 2.3.	Results for SQUAD V2 Dataset (Devlin et al. 2018)	8
Figure 2.4.	Deep Average Network (DAN) (Cer et al. 2018)	9
Figure 2.5.	Transformer Based Model of USE (Cer et al. 2018)	10
Figure 2.6.	DAN Based Model of USE (Iyyer et al. 2015)	11
Figure 2.7.	The Structure of the Skip-Thought Task (Kiros et al. 2015)	12
Figure 2.8.	The Structure of the Smart Reply Task (Yang et al. 2018)	12
Figure 2.9.	NLI Task (<i>The Stanford Natural Language Processing Group</i> 2022)	13
Figure 3.10.	Importing the Information from CSV to Dataframe	16
Figure 3.11.	Implementation of Cosine Similarity	16
Figure 3.12.	Loading the USE Module with Tensorflow	17
Figure 3.13.	Example Usage of DAN based Universal Sentence Encoder	18
Figure 3.14.	Example of Scoring a Grade	18
Figure 3.15.	Example of Encoded and Original Grades	19
Figure 3.16.	Git Repository	21
Figure 4.17.	Correlation of Proposed Models	24
Figure 4.18.	Comparison Graph of USE and Original Grades	25
Figure 4.19.	Comparison Graph of RoBERTa-Large and Original Grades	26
Figure 4.20.	Comparison Graph of BERT-Base and Original Grades	27
Figure 4.21.	Comparison Graph of Ensemble and Original Grades	28
Figure 4.22.	Original Distribution of Grades	30
Figure 4.23.	BERT-Base Distribution of Grades	31
Figure 4.24.	RoBERTa-Large Distribution of Grades	32
Figure 4.25.	USE Distribution of Grades	33
Figure 4.26.	Ensemble Distribution of Grades	34
Figure 4.27.	Z-Score Distribution of Original Grades	35
Figure 4.28.	Z-Score Distribution of DAN based USE Grades	36

List of Tables

Table 4.1.	Results Table	29
-------------------	-------------------------	----



1. INTRODUCTION

With the development of computer technology, its use and importance in the field of education has increased considerably. Leading the use of computers in online education and providing the ability to assess the students performances in many ways. Natural language processing and machine learning applications play a key role in the assessment process. Worldwide known exams TOEFL, GMAT, GRE are great examples that use the applications of natural language processing and machine learning. These computerized testing systems are reducing time and money spent on examination and implicitly helping to protect the ecosystem by not using papers in the process hence prevent the cutting of trees.

Besides the advantages mentioned above, these systems also provide the instructor to assess the performance of each student in detail. Assessment of the performance of the students can be done in many ways. Applying quizzes, exams is the most common way to observe the ability level of the students whether it is applied on pen and paper based or computer based. In order to fully assess the ability level of the student, answer of the student should be analyzed in a way to cover all the aspects of the sentence. Aspects can be divided as:

- Structure Based: Compare the structure of the student's answer to solution.
- Word Based: Compare the word similarity between the student's answer and the solution.
- Vector Based: Transform both student's answer and the solution into vectors and compare the similarity of the vectors.

The aspects mentioned above are each an area of expertise and when combined together can provide the best result. However, in the scope of this thesis, each aspect is analyzed individually and results taken from each aspect are compared in order to determine which aspect is suggested to have more likely results with the instructor's assessment.

Structure based assessment process is found not optimal because there are many grammar errors, misspelling, use of other languages (Turkish) than English are observed which can be considered normal due to anxiety in exam since most of the students attending the

exams are Turkish and English is not their native language. Hence assessment of the structure of the answers fail in many examples and sometimes even the irrelevant answers are shown as similar due to similarity in word order.

Word based assessment process is also not found optimal because the solutions of the questions asked in the exam differs in number of words and in some examples solution is simply made of a keyword. If only word similarity or even synonym similarity is applied for the assessment, an irrelevant answer containing only the required words may have high grade which is not the desired result.

Vector based assessment is found more suitable for sole assessment technique. Process of the vector based assessment can be summarized as:

- Take the sentences and transform them into high dimensional vectors that can be used for semantic similarity, text classification and other natural language tasks.
- Compare the vectors of student's answer and the solution to find similarity between them.

2. LITERATURE REVIEW

2.1. Computer Based Assessment

CBA(Computer Based Assessment) is the technique that is used to assess the students with the help of computer environment. It is often confused with the of CAA(Computer Assisted Assessment) but in CBA exam and grading is delivered by the computer where in CAA only the exam is made by the computer and grading is done by the instructors. This detail is what makes CBA different compared to CAA. As stated in introduction, worldwide known exams like TOEFL, GMAT, GRE are examples that is considered as CBA exams. There are many advantages of using CBA over classic paper based exams. Advantages are as following:

- Immediate score reporting: According to the study Daniels and Gierl 2017, anxiety and anger of the students are dramatically reduced when they learn the their grades immediately after the exam is finished.
- Time Efficiency: Computer based assessment is a fast process thanks to the high technology. Reading papers takes long time and requires high effort to evaluate each paper carefully. With CBA, papers are evaluated immediately after the exam is finished which prevents the effort of reading papers and saves huge time to give feedback on the students about the mistakes, misunderstandings, areas to be improved etc.
- Storing Space: Paper based exams require storing the papers in real world environment and covers too much space and exams should be preserved carefully to prevent crisis. However, CBA does not have this kind of the problem because the data of the exams are stored in hard drives or servers meaning they cover minimum physical space and easy to preserve and backup.

Advantages of the CBA over paper based exams have been discussed but there are also disadvantages of using CBA. Disadvantages are as following:

- Cost: CBA exams requires computers or supported devices such as smartphones, tablets etc. In order to make the systems on computerized environment, it is needed

to have a platform developed using a programming language and deploy the program into an environment where anyone attending exam are able to reach. This process is expensive in both hardware cost and the deployment part. Although the cost is high for the beginning, in long term investment may pay off compared to the paper based exams.

- **User Interaction:** Although in present time almost every human in the world has access to internet and computerized environment, it is hard to design a generic application that will be easy for users with low experience using computers. There may also be bugs, errors, operating system or web browser incompatibility.
- **Availability:** CBA should be accessible to each student meaning each student should have access to internet in case of a remote examination or even a computer to attend the exam. It may be rare to not have a computer in today's conditions however considering the financial status of the students there may be some students that are not able to participate in online exams. So if the exam is decided to be made on CBA system, availability for all of the students should have been provided first.

2.2. Natural Language Processing

Natural language processing (NLP) is a field of computer science which provides technology to understand the interaction between human and computers with the help of artificial intelligence by analyzing large amounts of natural language data. The aim is understand the contents of documents, including the contextual differences of the specific language within them. Using NLP one is able to (Natural Language Processing(NLP) 2021):

- **Sentiment Analysis:** In order to understand whether a given document is positive, negative or neutral to analyze the data of customers, students, companies etc. By applying sentiment analysis, it is possible to create a marketing plan by reviewing the feedback and improve quality of the services given.
- **Named Entity Recognition:** NLP can be used to detect names, organizations, locations, financial values etc. to have detailed analysis on the relations between the entities in the document which leads to better understanding of the discussion.

- **Text Summary:** Given a long document, NLP is able to summarize the text of the document into a meaningful paragraph and may save time to summarize the document for presentation or preparing the document for a report.
- **Topic Classification:** As mentioned in named entity recognition, NLP is able to detect the topics inside the document which is suitable for classifying the topics and create a database for further use. It is also advantageous to have information about the topics with little effort for time saving.
- **Text Cleaning:** It is possible to convert raw text data taken from customers, students etc. into structured and understandable format using NLP.

2.2.1. NLTK

NLTK is a platform to build Python applications to study language data. It is easy-to-use for over 50 corpora and lexical resources such as WordNet, with many text processing libraries for classification, tokenization, stemming, tagging etc (Madnani 2007).

2.2.2. TF-IDF

TF-IDF(Term Frequency-Inverse Document Frequency) is used in information retrieval and machine learning to detect the importance of a document in a dataset. Basically TF-IDF counts the words used in a given document to detect the importance of the specific word in the document. By checking the frequency of the word in the document, TF-IDF is able to recommend the popular terms in a subject hence it is widely used in search engines and movie, song recommender systems. Formulation for term frequency is given in 2.1 and formulation for inverse term frequency is given in 2.2.

$$tf(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad (2.1)$$

$$idf(t, D) = \log \frac{N}{|\{d \in D : t \in d\}|} \quad (2.2)$$

By combining the equations 2.1 and 2.2 TF-IDF is calculated as shown in equation 2.3.

$$tfidf(t, d, D) = tf(t, d) \cdot idf(t, D) \quad (2.3)$$

2.2.3. T5

T5(Text-to-Text Transfer Transformer) model is proposed by the study(Raffel et al. 2020) in 2020. In this approach, each NLP problem is treated as text-to-text problem. To be able to use this approach in different tasks, while the sequence of the process is done, a unique prefix that tags the text for a specific task is added to the input sequence. In the Figure 2.1, diagram of the model is shown.

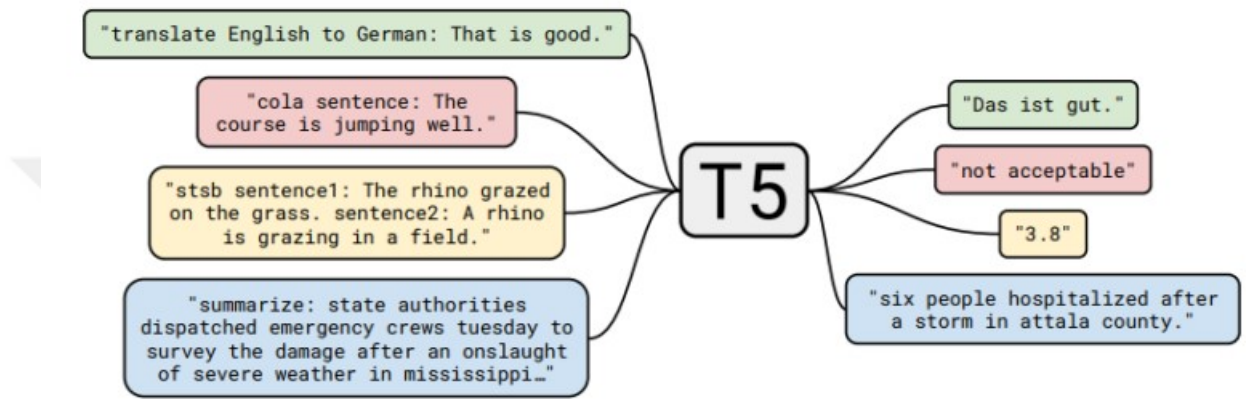


Figure 2.1. Diagram of the text-to-text framework (Haller 2020)

As it can be seen in the diagram, aim of this approach is to have a all in one model for all the natural language processing tasks. For the most common datasets like SciEnts-Bank, model performed a state-of-art performance. However, for the university context, result were lower as expected since the data size and complexity of the analyzed documents are different.

2.2.4. Transformers

Transformers are the deep learning models that proposed in the paper (Vaswani et al. 2017) and are widely used in natural language processing and computer vision tasks. Before transformer models, RNN(Recurrent Neural Networks) was the most popular technique in the field of natural language processing. However, RNN was not able to process the entire input at the same time. As stated in the study (Vaswani et al. 2017), an attention mechanism was proposed to provide parallelization in the process and enabled the effective analysis of a given sentence by processing the whole sentence.

Functions mentioned above made it possible to create pre-trained models on larger datasets in shorter times and fine-tuned if there is a special field of study. Some of the most popular pre-trained transformer models are BERT (Bidirectional Transformer Encoder Representations from Transformers), GPT (Generative Pre-trained Transformer) and USE (Universal Sentence Encoder). BERT and USE are mostly used in natural language processing tasks but GPT is mostly used as an artificial intelligence model for wide variety of tasks.

Representation of a transformer model is shown in Figure 2.2 and key point in different models that using transformer architecture is generally the number of layers that is been used in the process. If there is no need for very detailed analysis, selecting the models that use lower number of layers will save time and computation power.

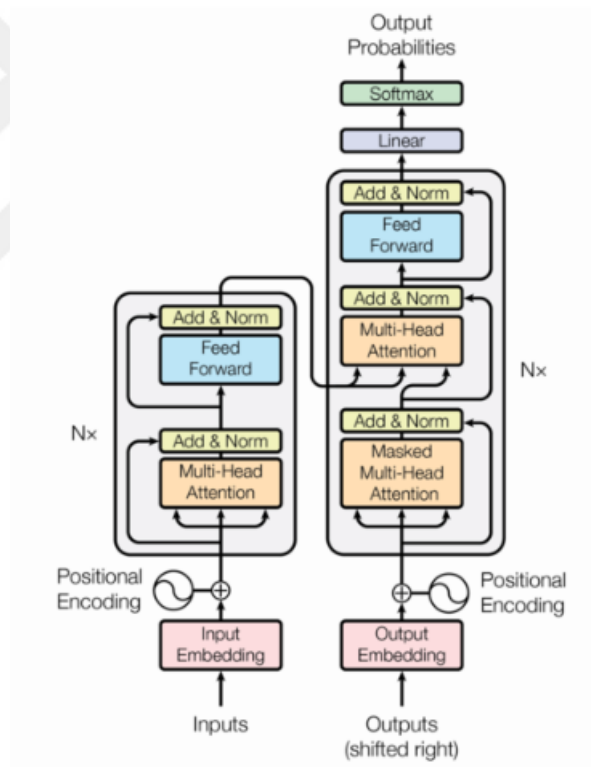


Figure 2.2. Representation of Transformer (Vaswani et al. 2017)

2.2.5. BERT

Bidirectional Encoder Representations from Transformers (BERT) is a transformer-based machine learning technique for natural language processing (NLP) pre-training de-

veloped by Google. It was created and by Jacob Devlin and his colleagues in 2018 and published in 2019 (Devlin et al. 2018). Previous models developed for natural language processing were using unidirectional language models which can only reach either left or right of the evaluated tokens. To deal with the limitations of this unidirectional model, in the paper (Devlin et al. 2018), it is shown that when applied to the most popular natural language processing datasets like (The General Language Understanding Evaluation), SQUAD (The Stanford Question Answering Dataset) and SWAG (The Situations With Adversarial Generations), demonstrated a state of art quality performance for natural language processing tasks. Example results for SQUAD dataset is shown in Figure 2.3.

Tasks	Dev Set				
	MNLI-m (Acc)	QNLI (Acc)	MRPC (Acc)	SST-2 (Acc)	SQuAD (F1)
BERT _{BASE}	84.4	88.4	86.7	92.7	88.5
No NSP	83.9	84.9	86.5	92.6	87.9
LTR & No NSP	82.1	84.3	77.5	92.1	77.8
+ BiLSTM	82.1	84.1	75.7	91.6	84.9

Figure 2.3. Results for SQUAD V2 Dataset (Devlin et al. 2018)

As shown in the Figure 2.3, BERT outperformed its opponents and created the basis for improvements in the field of natural language processing. BERT has many versions that is used for different tasks such as question answering, measuring sentence similarity, next sentence prediction, text classification, feature extraction etc.

2.2.6. Universal Sentence Encoder

The Universal Sentence Encoder (USE) is proposed in (Cer et al. 2018) and is used for encoding sentences into vectors to be analyzed for NLP tasks. There are currently two different versions of USE which differs in accuracy and performance. One is transformers based and the other one is based on Deep Averaging Networks (DAN). According to the paper (Cer et al. 2018), transformers based model has better accuracy but takes longer time and needs more resources. DAN based model is less effective dealing with large amounts of data compared to the transformers based model but needs little resources and still provides meaningful results when the data size is small. Hence for the projects that re-

quire less detailed analysis, DAN variant is the better choice. For the projects that require detailed analysis transformers variant is suggested. Although the hardware requirements change according to the model, in some cases transformer model uses less resources depending on the sentence length because it only needs to store unigram embeddings for the processing.

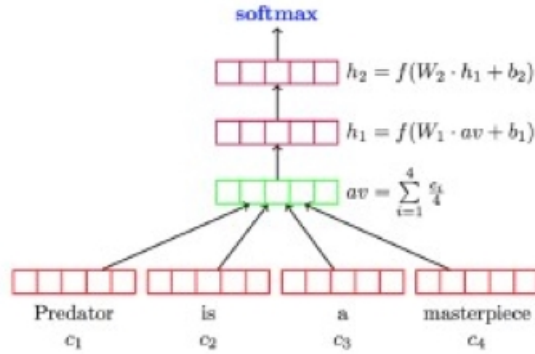


Figure 2.4. Deep Average Network (DAN) (Cer et al. 2018)

USE, BERT or transformer models in general have been developed in order to have a stabilized way of processing the natural language processing tasks in sentence level. Not using sentence level encoders may result in following problems:

- **Loss of information:** When using word level encoding methods, the important thing is how many words match between those two compared sentences. "It will be rainy tomorrow" and "It will be" is going to have a high similarity rating even if the two sentences are not similar in the sense of meaning.
- **Order of the words:** In natural language processing, it is important to understand whether the processed sentence is a question or a normal sentence. Word level encoding methods will provide high similarity rates between "It is a dog." and "Is it a dog." sentences. Even if one is a statement and the other one is a question, word level encoders will not detect the difference.

Using TF-IDF or other techniques that adds these capabilities may solve the problem but there is no need to put extra effort where there are models that are capable of doing higher level tasks with ease. As stated before, there are two variants for USE.

First one is transformer based and the second one is DAN based.

Transformer Based: Transformer based model of USE is composed of 6 stacked transformer layer with each layer having a self-attention module that provides an output of a 512-dimensional vector as sentence embedding. An example of the structure of the transformer based model is shown in the Figure 2.5

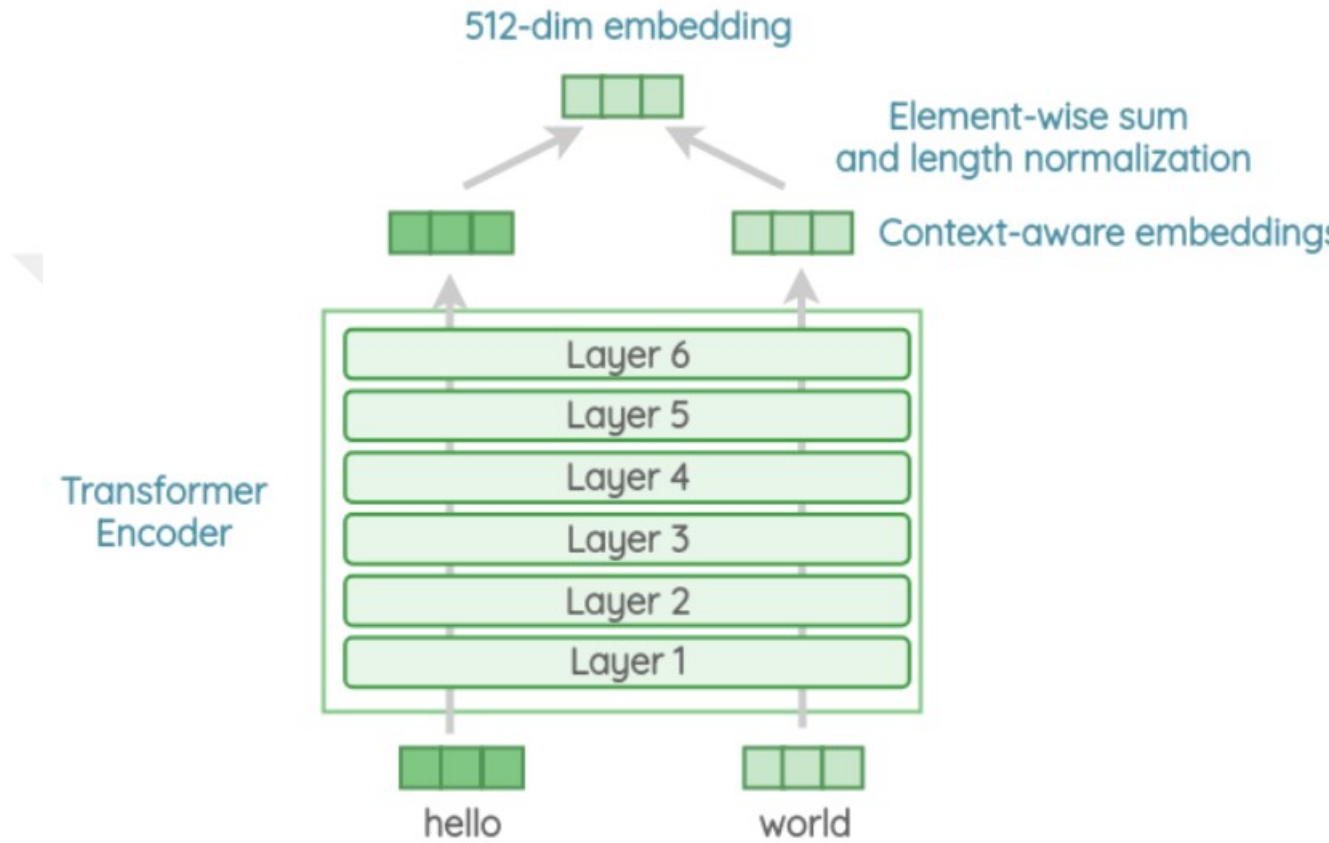


Figure 2.5. Transformer Based Model of USE (Cer et al. 2018)

Advantage of this variant is as mentioned before providing better accuracy but high resource consumption because of the complex architecture. Self-attention mechanism has $O(n^2)$ time complexity so longer sentences will have longer processing times.

Second one is the DAN based model which is considered simpler compared to the transformer based model. The architecture of this model is proposed in (Iyyer et al. 2015) and shown in Figure 2.6

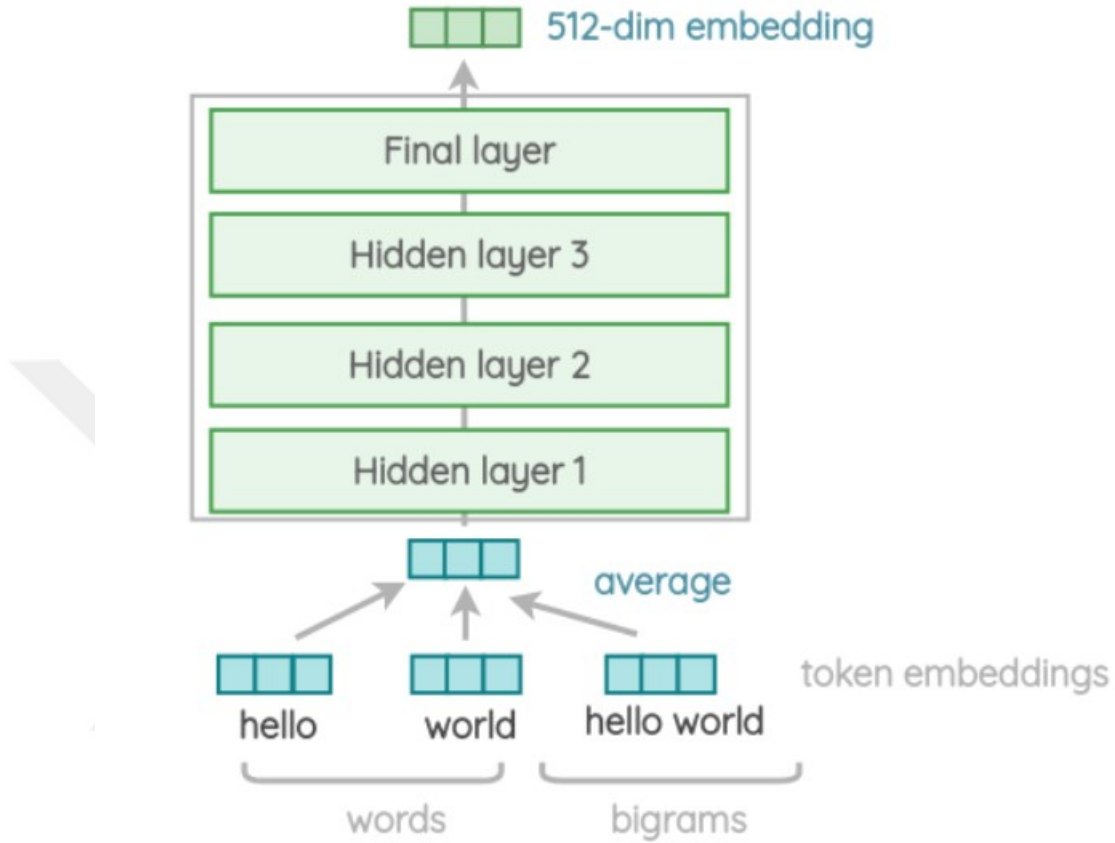


Figure 2.6. DAN Based Model of USE (Iyyer et al. 2015)

This model is composed of 4-layered feed forward DNN that provides an output of 512-dimensional vector as sentence embedding. Both models have been pre-trained on multitask learning tasks as following:

- **Modified Skip Thought:** The idea is proposed in the study (Kiros et al. 2015) to use the current sentence in a document to predict the previous and next sentence. Structure of the mechanism is shown in the Figure 2.7.
- **Response Prediction Task:** In this learning task, it is needed to predict the correct response from the given list of responses. Idea is proposed by the study (Yang et al. 2018) and being commonly used in the smart reply features in the mail providers. Structure of the smart reply mechanism is shown in Figure 2.8.

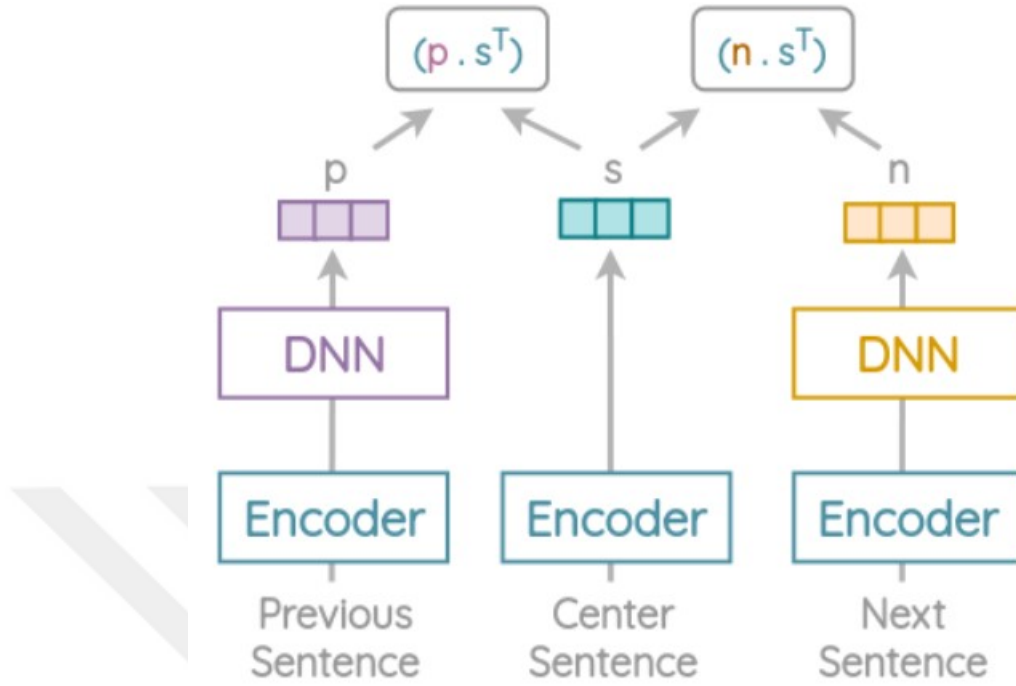


Figure 2.7. The Structure of the Skip-Thought Task (Kiros et al. 2015)

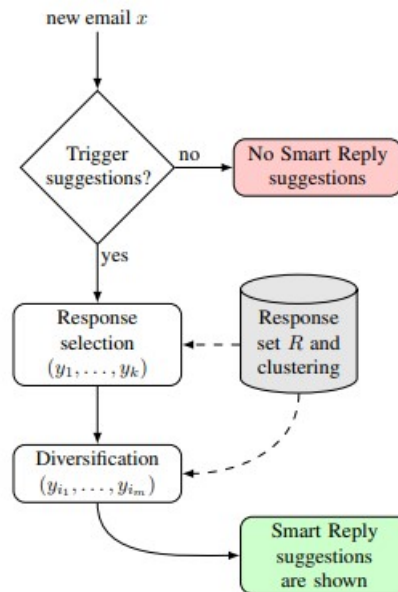


Figure 2.8. The Structure of the Smart Reply Task (Yang et al. 2018)

- Natural Language Inference Task: This task is to predict if there is a comparison of the availability of the hypothesis. In this task, SNLI(Stanford Natural Language Inference) Corpus(*The Stanford Natural Language Processing Group 2022*) is used for training. Aim of this task is to detect the relation between the hypothesis and the premise. There are three possible outputs for this task: entailment, contradiction or neutral. The structure of this task is shown in Figure 2.9. Using this structure, hypothesis testing is done and both models for USE are ready to use for many natural language processing tasks such as text classification, smart reply, clustering etc.

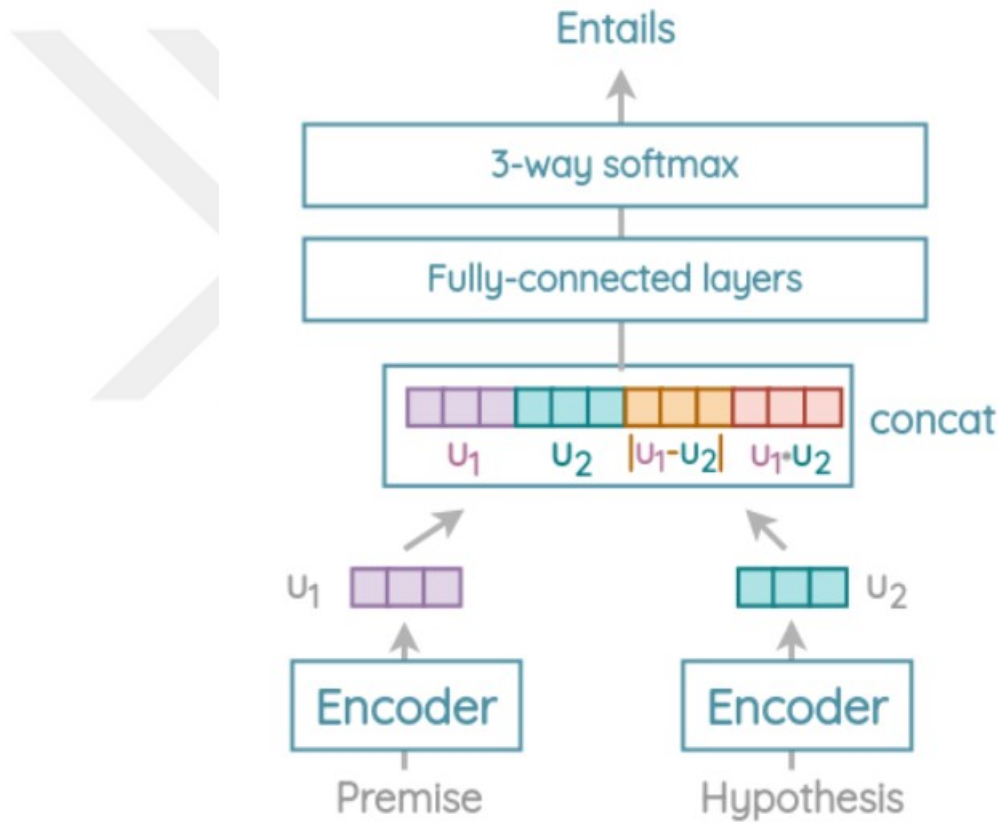


Figure 2.9. NLI Task (*The Stanford Natural Language Processing Group 2022*)

2.3. Similarity Measures

2.3.1. Cosine Similarity

Cosine similarity technique is the preferred technique for scoring the grades in this study because documents are represent as vectors in this technique and angle between two vectors determine the similarity rate of two sentences. In other words, by transforming each student answer sentence into the integer array vectors and checking the angle of two compared sentences, similarity can be mathematically measured. Formula for cosine similarity is given in 2.4.

$$\text{Cosine}(x, y) = \frac{x \cdot y}{|x||y|} \quad (2.4)$$

2.3.2. Z-Score

Z-Score is a metric that is used to determine how far a data point in the dataframe is from the mean. In order to detect how many of the original and graded scores are in the same range, by using Z-Score, both grades are standardized and can be compared in terms of the accuracy. After calculating the final results, Z-Score is required to observe how accurate was the evaluation process. Formula for Z-Score is given in 2.5.

$$z = \frac{x - \mu}{\sigma} \quad (2.5)$$

3. MATERIAL AND METHOD

In the scope of this thesis, a Python module to automatically grade the open ended answer of the students is developed. Python programming language is selected because natural language processing libraries are easy to import and use. In this section, process of scoring student answers of the exams of Natural Sciences course were explained.

3.1. Python Programming Language

Python is a general purpose programming language that supports multiple programming paradigms such as object-oriented and functional programming with emphasis on code readability and indentation *A Python Book 2012*.

Python is widely used in the industry in the areas of machine learning, automation, web scraping, deep learning, computer vision etc. Python is a good choice for this study because it is easy to install and start the development process in almost any operating system with various libraries that are available and good documentation for each of these libraries. There are many code editors that has built-in support for Python and for this study Visual Studio Code is preferred as the programming environment.

3.2. Visual Studio Code

Visual Studio Code is a text editor with great customization options due to its support for community extensions. As in most popular text editors such as Sublime Text, Atom etc. it is possible to change the theme, fonts, sizes of the text and increase the readability of the code but what makes Visual Studio Code a great environment to use for programming is that it is possible to turn it into a compiler for different programming languages using community extensions. For this study, Jupyter Notebook extension of the Visual Studio Code is used.

3.3. Pandas Library

Pandas is a data analysis library for Python programming language. Supporting many file extensions such as CSV(Comma Separated Values), Excel, JSON(JavaScript Object Notation), pandas allows the user to import the data stored from the mentioned file formats into dataframes for tasks such as cleaning, analyzing, mining. In this study, data cleaning and analysis is made by importing the exam information from the CSV files into dataframe. Example of importing the information into dataframe for one question is shown in Figure 3.10

```
examQ1 = pd.read_excel("NSexam.xlsx", sheet_name = 'Q1')
examQ1.fillna('', inplace=True)
examQ1.head()
answersQ1 = examQ1.iloc[:,2].values.tolist()
originalgrades = examQ1.iloc[:,5].values.tolist()
```

Figure 3.10. Importing the Information from CSV to Dataframe

3.4. Numpy Library

Numpy is a library for Python programming language which adds support for high-level mathematical functions and multi-dimensional arrays and matrices Harris et al. 2020. In this study, to calculate the cosine similarity, linear algebra functions built-in from the Numpy library is used. Implementation of the cosine similarity function is shown in Figure 3.11 .

```
def cosine(u, v):
    return np.dot(u, v) / (np.linalg.norm(u) * np.linalg.norm(v))
```

Figure 3.11. Implementation of Cosine Similarity

3.5. Scipy Library

Scipy is a library for Python programming language which provides scientific algorithms, equations and many other solutions for scientific problems. In this study, Scipy is used to check the score of Independent t-test by importing the "stats" module.

3.6. Tensorflow

Tensorflow is an open source library for machine learning and artificial intelligence tasks. Developed by GoogleAbadi et al. 2016 in 2015. It is possible to use tensorflow in many programming languages such as Python, C++, Javascript. Using tensorflow's hub module, it is easy to import a pre-trained module into the program to be used in the development. Example usage of the module loading and output is shown in Figure 3.12

```
module_url = "./universal-sentence-encoder_4"  
model = hub.load(module_url)
```

Figure 3.12. Loading the USE Module with Tensorflow

3.7. BERT

BERT is pre-trained transformer model that is widely used in natural language processing. Some features of BERT include:

- Question Answering: BERT is able to answer questions from a given reference document and accurately give correct responses.
- Summarization: BERT is able to summarize a given document with the fine-tuned models.
- Sentence Similarity: With many models pre-trained on different large datasets mentioned before, using cosine similarity, it is possible to use a pre-trained BERT model to find similarity between two sentences.

In this study, pre-trained models "stsb-roberta-large" and "sentence-transformers/bert-base-nli-mean-tokens" are used as the alternative of USE for comparing the results. These models were selected because of the high performance observed in different datasetsDevlin et al. 2018.

3.8. Universal Sentence Encoder

USE is a model explained in literature review part. It has two different variations for encoding sentences into vectors. First one is the transformer architecture based and the second one is the DAN based model. In this study, DAN based model is preferred because the sample size is low. Usage of the model is simple:

- Download the pre-trained model from tensorflow hub.
- Load the module using tensorflow.
- Apply the model to each answer in the exam file and for the solution manual.
- After the model is applied and all the sentences have been encoded as vectors, calculate cosine similarity between each sentence and provide corresponding scores for each student.

Example usage for one question in the code is shown in the Figure 3.13

```
answersQ1 = examQ1.iloc[:,2].values.tolist()
originalgrades = examQ1.iloc[:,5].values.tolist()
sentence_embeddings = model(answersQ1)
query = 'Due to internal crystal structure of the molecules.'
query_vec = model([query])[0]
```

Figure 3.13. Example Usage of DAN based Universal Sentence Encoder

After the process is completed, next step is to score each answer using cosine similarity and draw the plot for grade distribution. An example is shown in Figure 3.14.

```
for sent in answersQ1:
    sim = cosine(query_vec, model([sent])[0])
    grade = sim*20
    if grade<0:
        grade = 0
    data.append(round(grade))
    print("Sentence = ", sent, "; similarity = ", sim)
```

Figure 3.14. Example of Scoring a Grade

Number:	Encoder:	Original
20160807006	9 20	
20160807009	4 0	
20170808017	10 20	
20170808045	2 12	
20190808008	5 0	
20190808013	3 0	
20190808014	3 0	
20190808023	7 12	
20190808024	3 0	
20190808068	4 0	
20190808092	8 0	
20195156020	9 20	
20195175010	5 12	
20200808001	1 0	
20200808005	6 20	
20200808006	6 5	
20200808009	6 12	
20200808014	6 0	
20200808015	8 5	
20200808016	1 0	

Figure 3.15. Example of Encoded and Original Grades

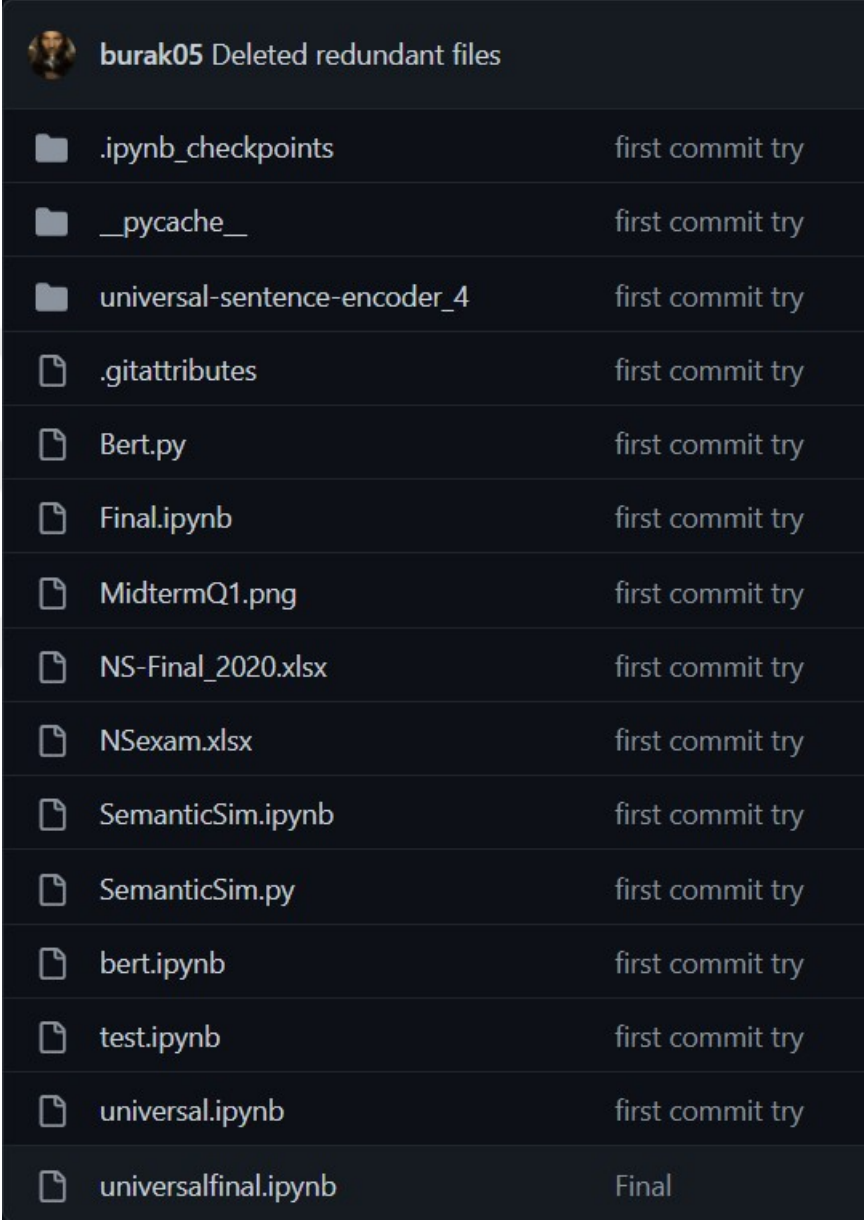
After the process mentioned in 3.14 is done, an example output is printed on the screen to compare the graded notes with original grades given by the instructor.

As shown in the example output in Figure 3.15, program scores are quite different compared to the original scores given by the instructor.

3.9. Git

Git(Global Information Tracker) is a open source version control system to manage projects with different sizes in a remote and effective environment. Git is commonly used in software engineering projects which requires collaboration and process tracking. With the branch system of Git, it is possible to have different versions of the project in order to provide diversity in the projects.

There are many websites that use Git repositories. Github and Gitlab are some of the most popular sites that use Git. In this study, in order to backup and track the development process, a repository on Github is created. In order to save computation time and continue the development offline, used models were needed to be uploaded on repository. Github's maximum file size upload limit is 50 MB and the USE model that is applied on the project exceeds that limit. Hence, to upload the larger files on the repository, Git LFS(Large File System) extension was the solution for that problem by handling the upload of the large files to the repository. Repository structure that is been used during the study is shown in the Figure 3.16



burak05 Deleted redundant files	
folder	.ipynb_checkpoints first commit try
folder	__pycache__ first commit try
folder	universal-sentence-encoder_4 first commit try
file	.gitattributes first commit try
file	Bert.py first commit try
file	Final.ipynb first commit try
file	MidtermQ1.png first commit try
file	NS-Final_2020.xlsx first commit try
file	NSexam.xlsx first commit try
file	SemanticSim.ipynb first commit try
file	SemanticSim.py first commit try
file	bert.ipynb first commit try
file	test.ipynb first commit try
file	universal.ipynb first commit try
file	universalfinal.ipynb Final

Figure 3.16. Git Repository

3.10. Computer Specifications

This thesis is developed in a computer with following specifications:

- Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.60 GHz
- 16 GB of RAM
- NVIDIA GTX 1660TI 6 GB Graphics Card
- Python 3.8 installed on the device
- Windows 11 Pro Operating System

4. RESULTS AND DISCUSSION

In this study, methods discussed in the previous sections were applied on the exam data of the Natural Sciences course. During the development of this thesis, many techniques for scoring the text-based open-ended questions were analyzed and pre-trained models for this task "RoBERTa" Liu et al. 2019, "bert-base-nli-mean-tokens" which is proposed in "Sentence-BERT" Reimers and Gurevych 2019 and USE(Universal Sentence Encoder) Cer et al. 2018 is selected. By taking the average of the sum of each model's results, an ensembled result is also analyzed in this study. From the selected models, best performing model was USE according to the Figure 4.17. Following the result of USE model is the Ensembled model's results and last one is BERT-Base model.

Looking at the p-values for each of the models, USE model has p-value of $0.0374 < 0.05$ meaning the results of USE model can be considered significant compared to the original grades given by the instructor. However, other models have p-values greater than 0.05 which means they are insignificant compared to the original grades.

All of the models scored a higher average compared to the original grades but scores given by the USE are the closest to the original grades. This is due to the variation of the USE model is DAN based in this study hence the performance is better for the small sample size.

Standard deviation value of the original grades is the highest compared to the other models. This is due to the fact that especially for BERT based models, it is problematic to calculate the similarity of empty sentences because empty sentences has " " character which tricks the model that there is a common object with the solution. However, USE model did not encounter such problems.

Comparison graphs of the selected models are shown in the Figure 4.18 for USE, Figure 4.19 for RoBERTa-Large and Figure 4.20 for BERT-Base model. As seen in the graphs, distribution of the grades are better in USE and performance decreases in the other models.

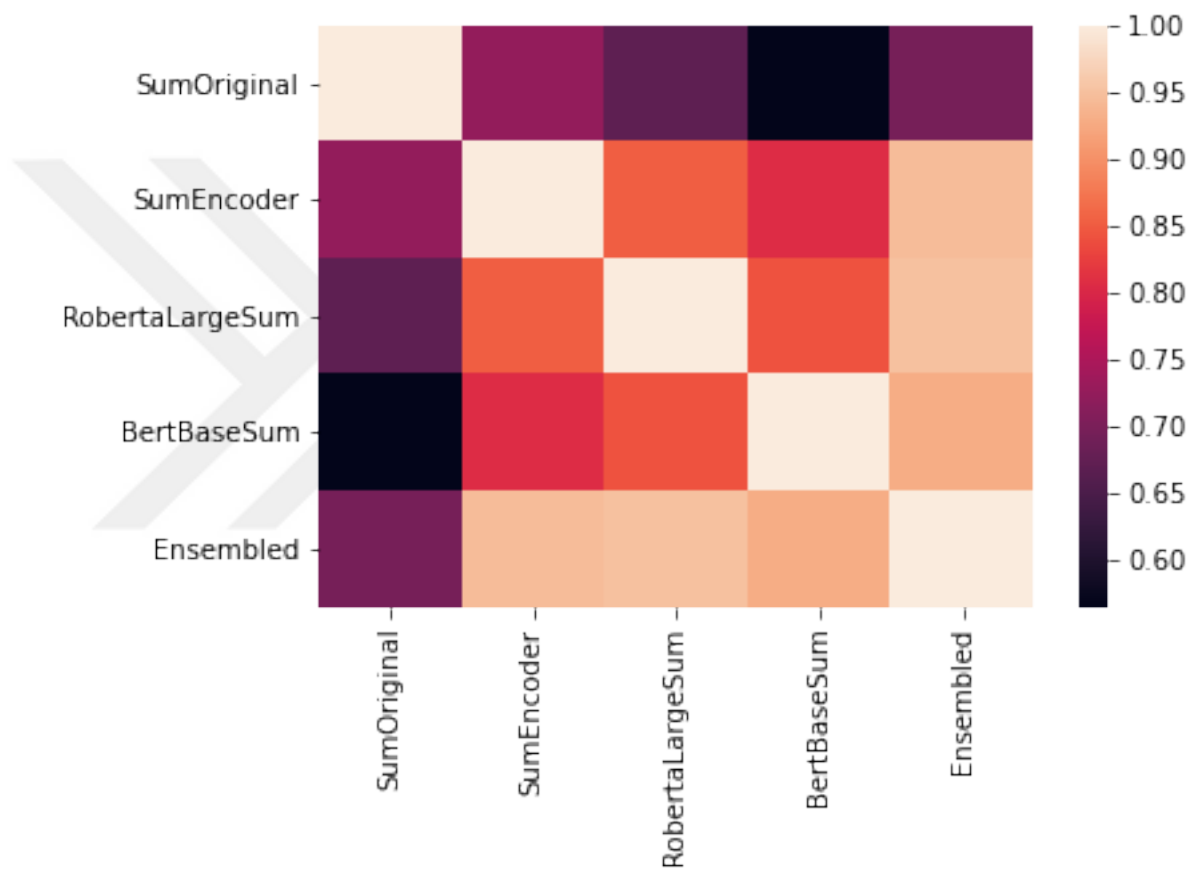


Figure 4.17. Correlation of Proposed Models

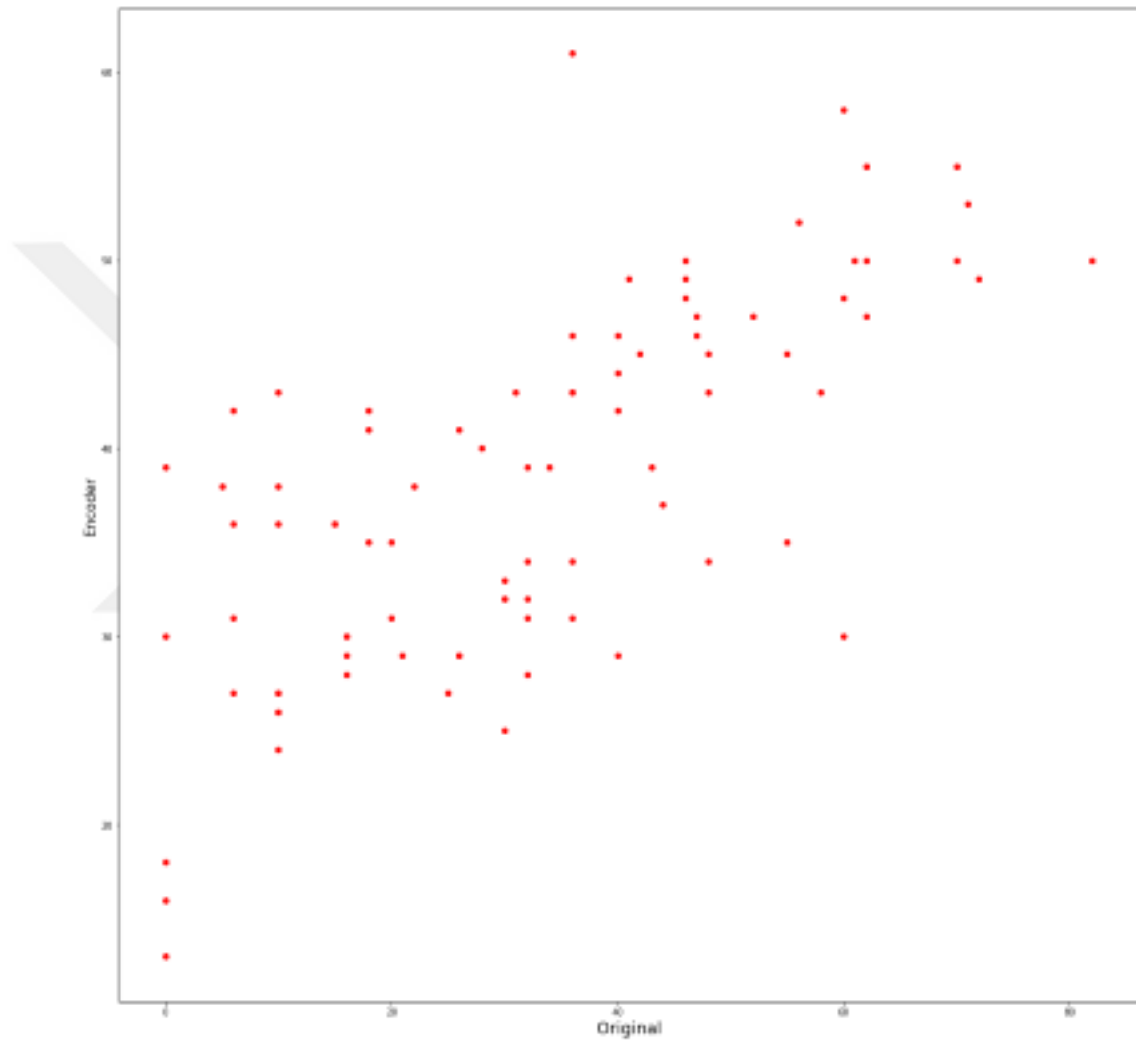


Figure 4.18. Comparison Graph of USE and Original Grades

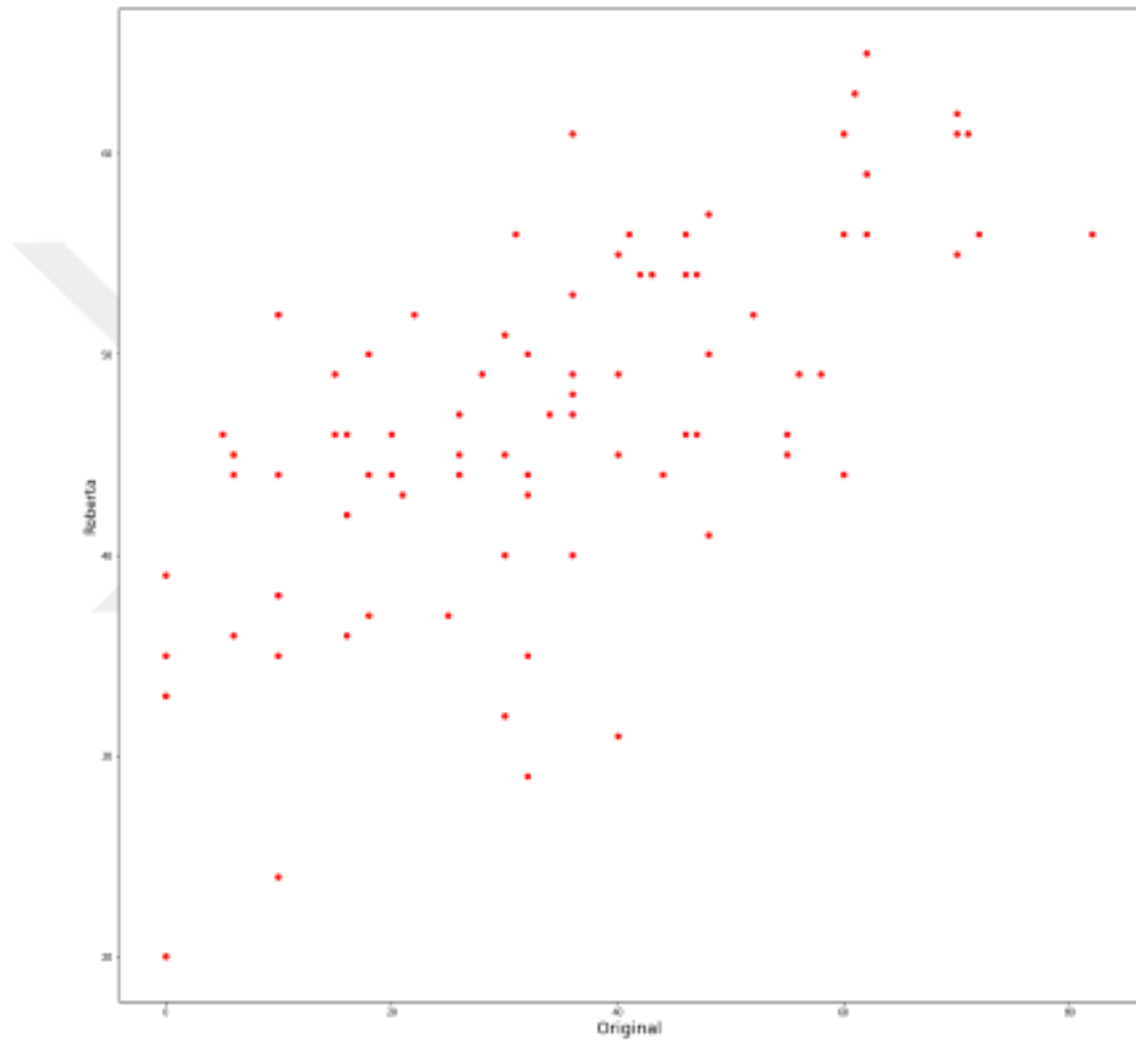


Figure 4.19. Comparison Graph of RoBERTa-Large and Original Grades

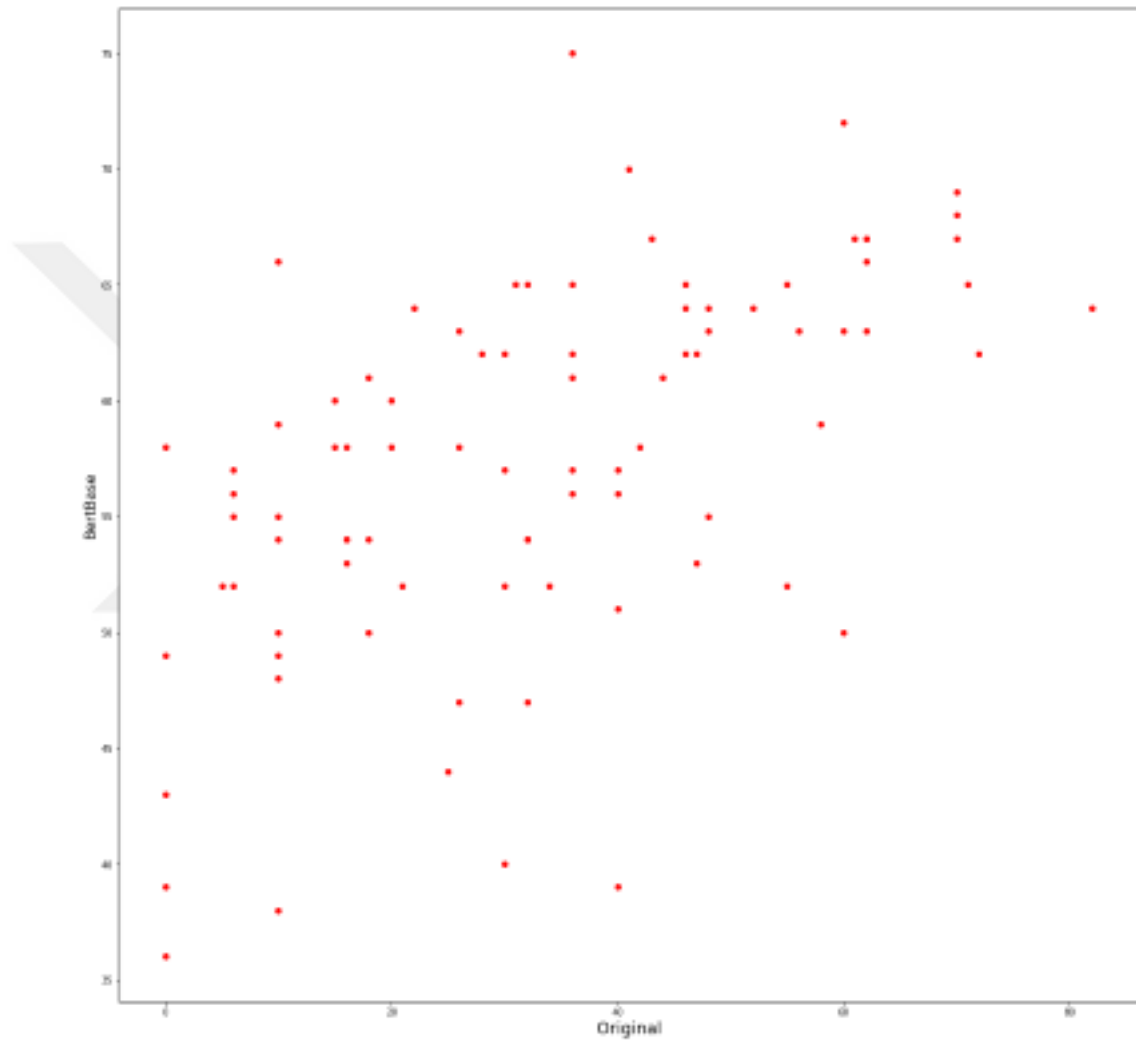


Figure 4.20. Comparison Graph of BERT-Base and Original Grades

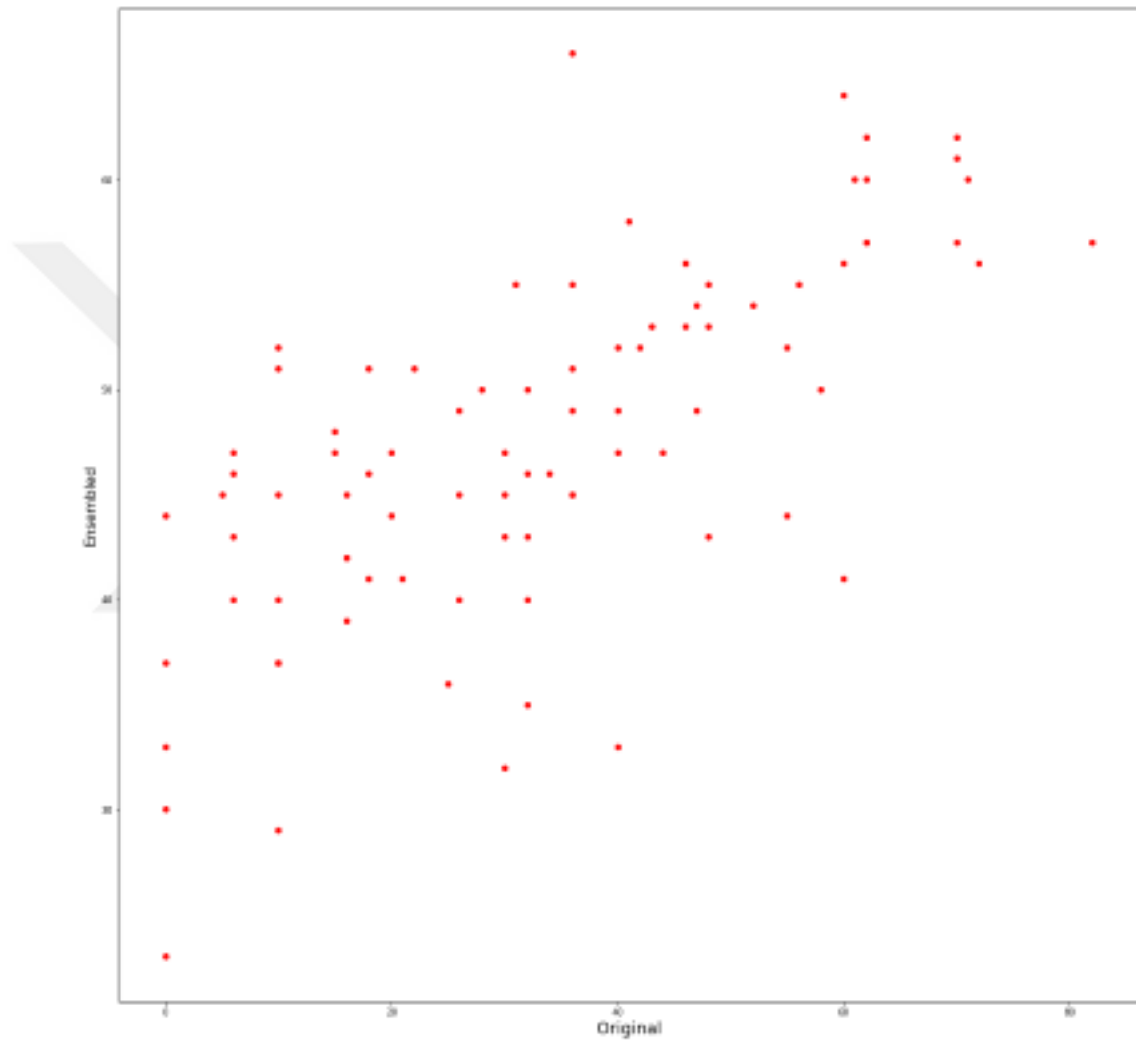


Figure 4.21. Comparison Graph of Ensemble and Original Grades

Table 4.1. Results Table

Grades	Mean	Standard Deviation
Original	33.386	20.68
USE	38.511	9.90
RoBERTa-Large	46.738	8.98
BERT-Base	57.545	8.03
Ensemble	47.590	8.51

After the mean and the standard deviation are calculated, first it is needed to observe the statistics of the students in terms of failed, successful and high success. Distribution of each models in terms of failed, successful and high success are shown in the following figures.

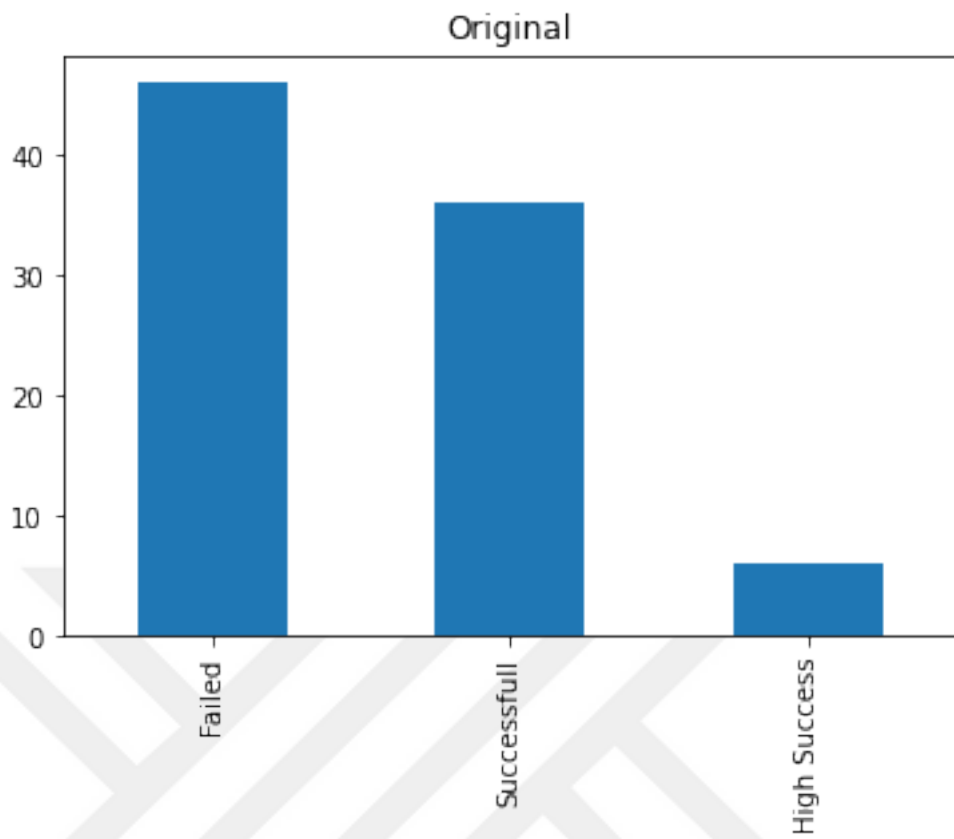


Figure 4.22. Original Distribution of Grades

In the original score distribution shown in 4.22, 46 students have Failed, 42 students were Successful and 6 of the passed students performed High Success.

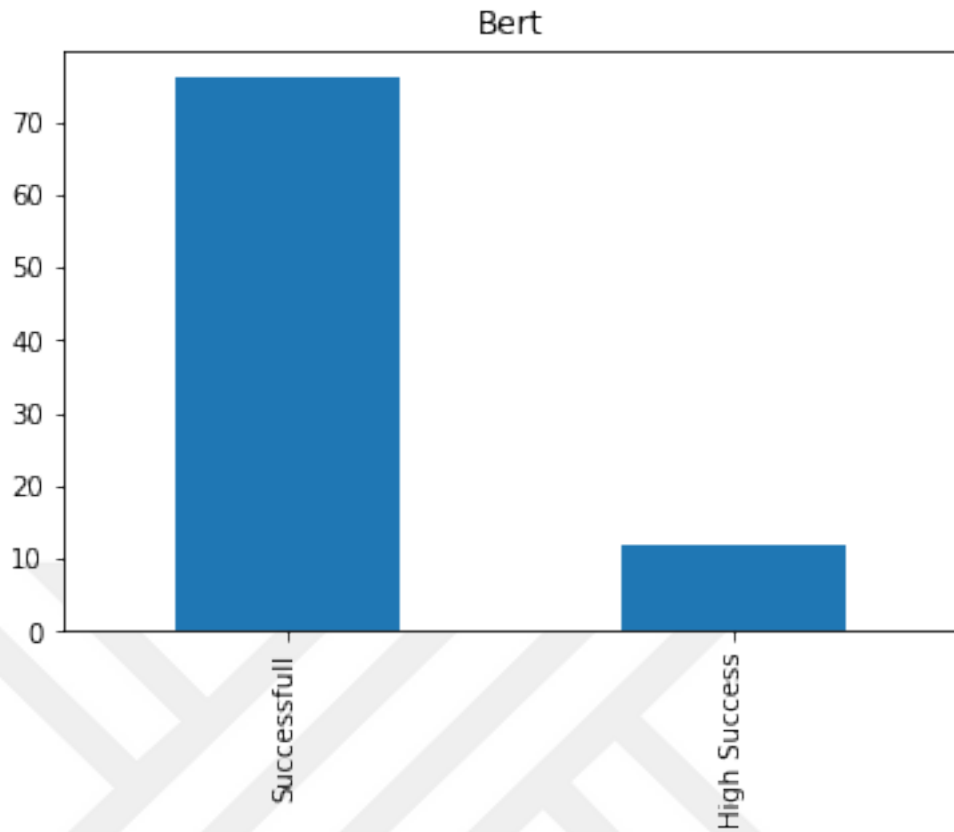


Figure 4.23. BERT-Base Distribution of Grades

Distribution of the grades assessed with BERT-Base model is shown in the Figure 4.23. In this model, no students have Failed and 12 students performed high success. This result is not viable because there are students that gave irrelevant answers to each question and should have failed the course.

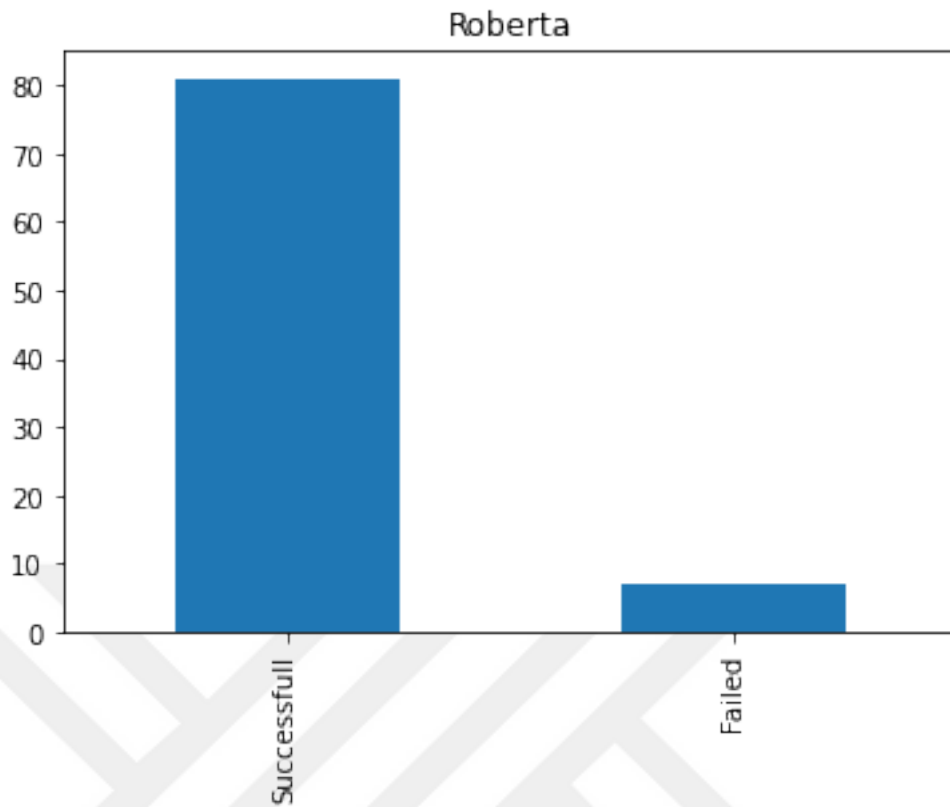


Figure 4.24. RoBERTa-Large Distribution of Grades

Distribution of the grades assessed with RoBERTa-Large model is shown in the Figure 4.24. In this model, 7 students have Failed and 81 students were Successful. This result is better than BERT-Base model but students with High Success in the original scores were given lower scores.

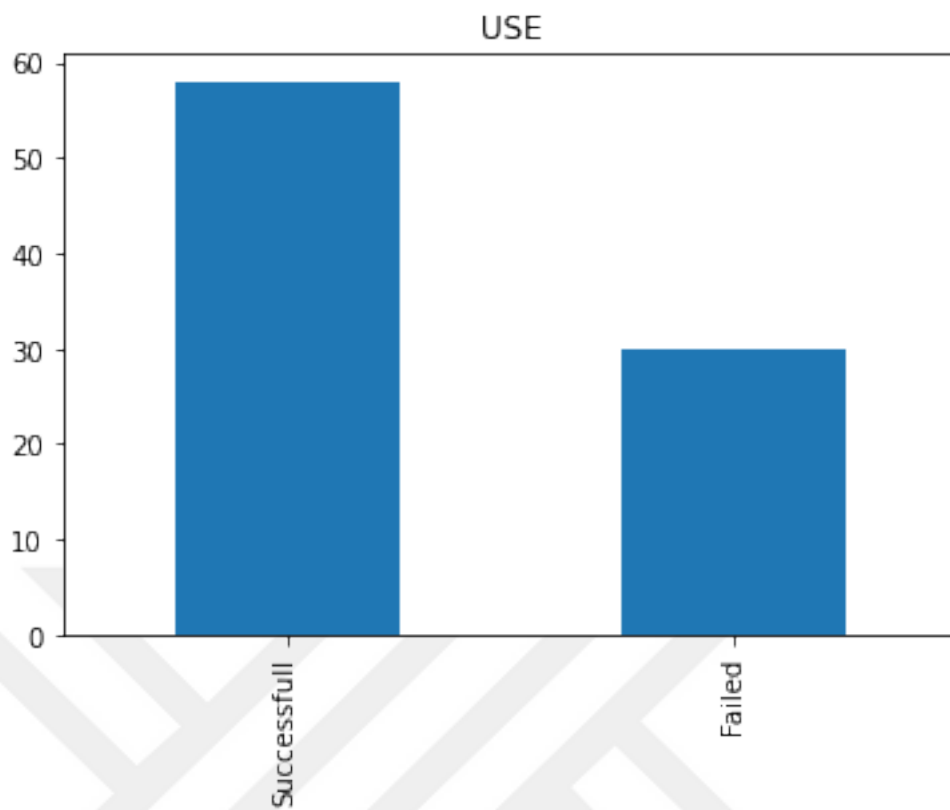


Figure 4.25. USE Distribution of Grades

Distribution of the grades assessed with DAN based USE model is shown in the Figure 4.25. In this model, 30 students have Failed and 58 students were Successful. There are no students that perform high success. This result is closest to the original scores given but students performed High Success in original assessment had lower scores.

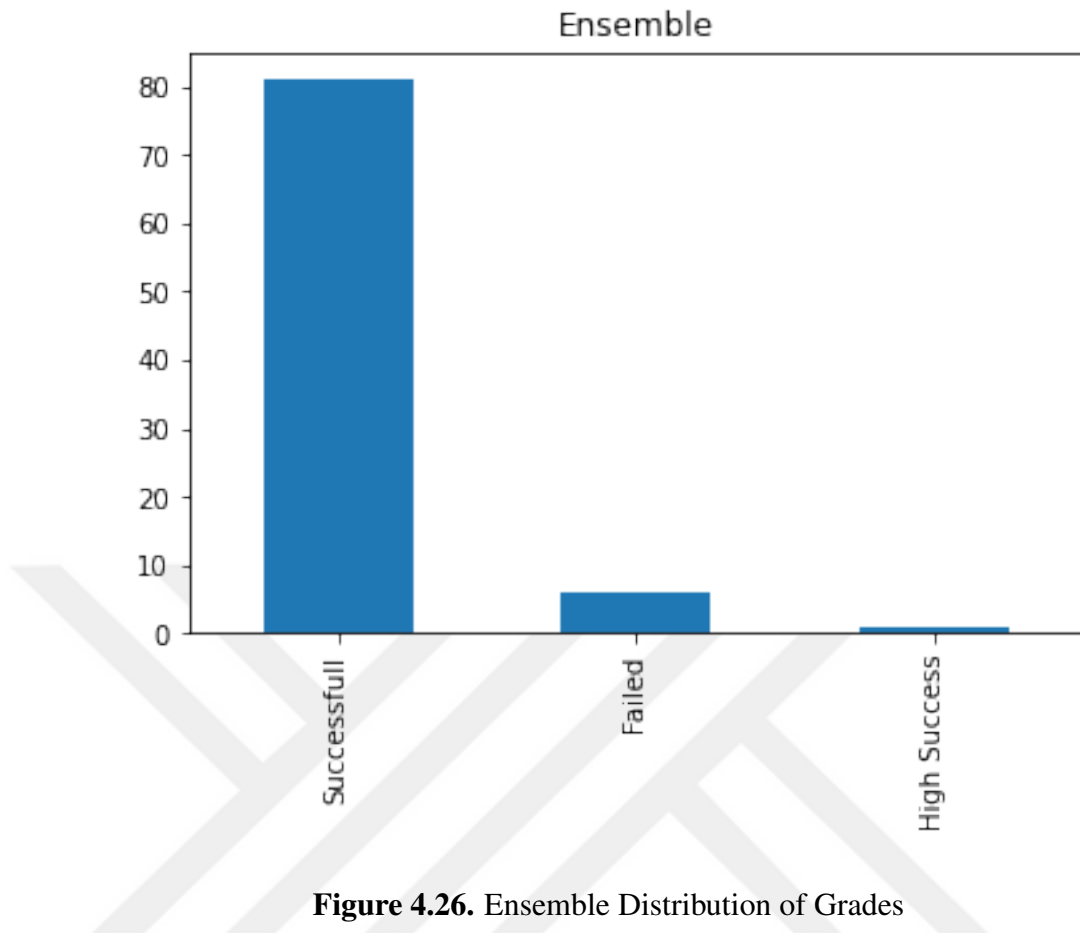


Figure 4.26. Ensemble Distribution of Grades

Distribution of the grades assessed with Ensemble model is shown in the Figure 4.25. Ensemble result is the average of the scores of used models in this study. In this result, 6 students have Failed and 82 students were Successful with 1 student performing High Success.

After the results were analyzed, it was clear that DAN based USE model is the best performing model overall. To observe the accuracy between USE model and original grades, Z-Score is applied on both grades.

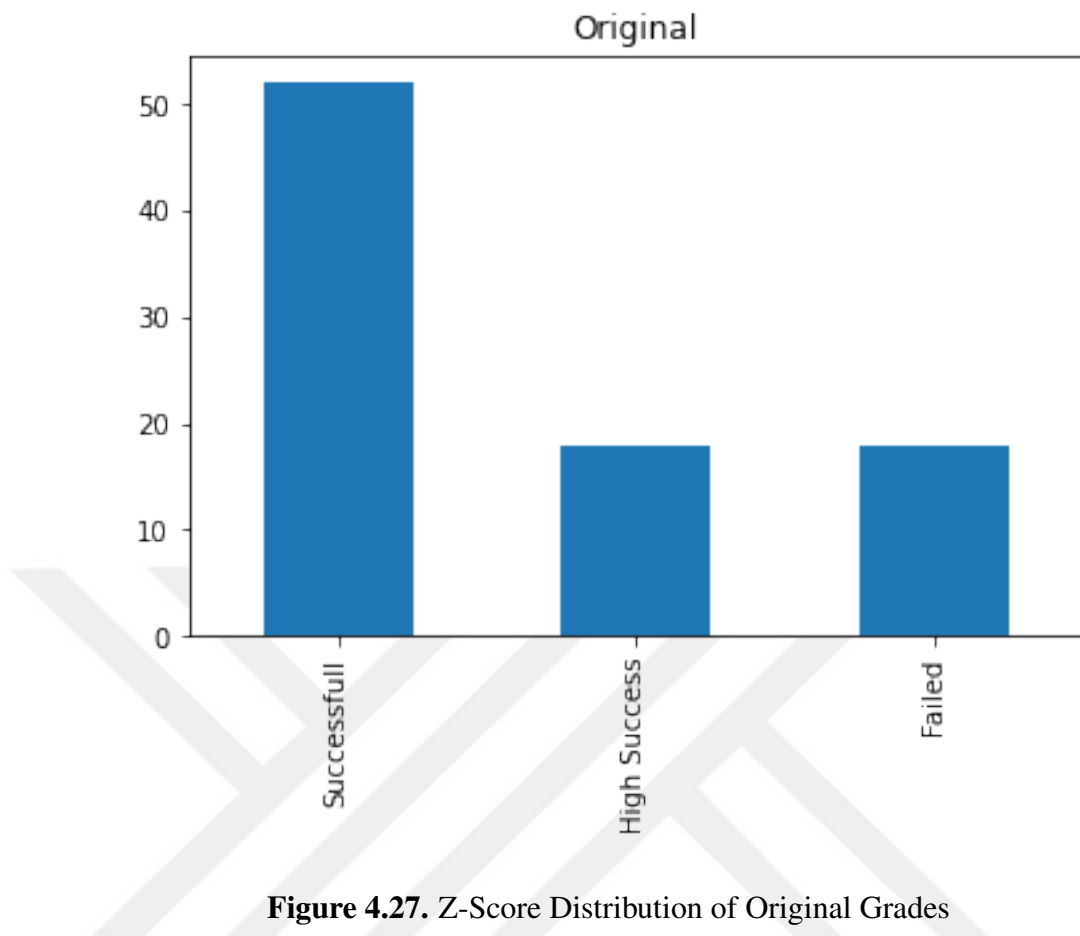


Figure 4.27. Z-Score Distribution of Original Grades

Distribution of the original grades after the Z-Score is applied is shown in the Figure 4.27. In this result, 18 students have Failed, 70 students were Successful and 18 of the passed students performed High Success.

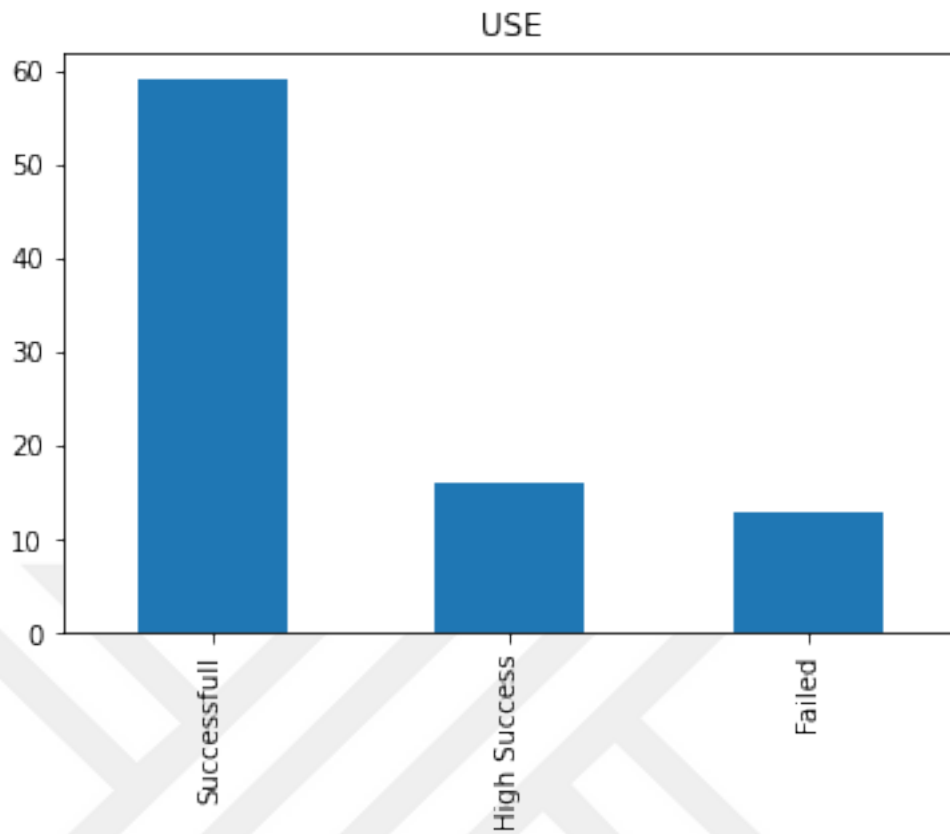


Figure 4.28. Z-Score Distribution of DAN based USE Grades

Distribution of the original grades after the Z-Score is applied is shown in the Figure 4.28. In this result, 13 students have Failed, 75 students were Successful and 16 of the passed students performed High Success.

By analyzing the data of Z-Score, it is observed that 65 of the students were labeled same in original and USE grades. From 88 students, 65 of the students were assessed similar to the instructor's assessment, meaning %74 accuracy is achieved by using the USE model in this study.

5. CONCLUSION

Within the scope of this thesis, question of if it is possible to evaluate the answers of text-based open-ended questions using vector based models of the variations of USE and BERT. Steps followed during this study are:

- After the literature review, mentioned models that are used in this thesis were selected.
- After the models have been selected, data was prepared for processing in Python programming language with the help of its large variety of data analysis libraries.
- Data is processed using three models: DAN based variation of USE, RoBERTa-Large and BERT-Base.
- After the processing, mean and standard deviation of each model are determined.

Summary of the findings in this study are as follows:

- In literature, evaluation of university level exams with custom datasets were missing. Each study is done on the popular large sized datasets.
- %74 accuracy is achieved with DAN based USE model.
- In this study, it is shown that using vector based models for evaluating the similarity of sentences in custom datasets with small size can be considered as a viable option.
- DAN based USE model outperformed other models used in this study in terms of comparison with the original grades and is suggested as the evaluation method for small sample size cases similar to this study.
- BERT based models suffer from having more complex structures and calculating the similarity of the sentence depending on how long is the sentence, hence they are inclined to give higher scores which is not optimal and not suggested to be used in similar studies.

As future work, this study is able to be improved by combining the models used and addition of adaptive capabilities in order to have optimized results which solves the problems encountered during this study is aimed.

6. REFERENCES

- Abadi, Martín et al. (Mar. 2016). “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems”. In: *arXiv:1603.04467 [cs]*. arXiv: 1603.04467. URL: <http://arxiv.org/abs/1603.04467> (visited on 05/25/2022).
- Cer, Daniel et al. (Apr. 2018). “Universal Sentence Encoder”. In: *arXiv:1803.11175 [cs]*. arXiv: 1803.11175. URL: <http://arxiv.org/abs/1803.11175> (visited on 05/19/2022).
- Daniels, Lia M. and Mark J. Gierl (Dec. 2017). “The impact of immediate test score reporting on university students’ achievement emotions in the context of computer-based multiple-choice exams”. en. In: *Learning and Instruction* 52, pp. 27–35. ISSN: 09594752. DOI: 10.1016/j.learninstruc.2017.04.001. URL: <https://linkinghub.elsevier.com/retrieve/pii/S0959475217302001> (visited on 05/23/2022).
- Devlin, Jacob et al. (2018). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: DOI: 10.48550/ARXIV.1810.04805. URL: <https://arxiv.org/abs/1810.04805> (visited on 05/19/2022).
- Haller, Stefan (Oct. 2020). *Automatic Short Answer Grading using Text-to-Text Transfer Transformer Model*. en. info:eu-repo/semantics/masterThesis. URL: <http://essay.utwente.nl/83879/> (visited on 05/25/2022).
- Harris, Charles R. et al. (Sept. 2020). “Array programming with NumPy”. en. In: *Nature* 585.7825, pp. 357–362. ISSN: 1476-4687. DOI: 10.1038/s41586-020-2649-2. URL: <https://www.nature.com/articles/s41586-020-2649-2> (visited on 05/23/2022).
- Iyyer, Mohit et al. (July 2015). “Deep Unordered Composition Rivals Syntactic Methods for Text Classification”. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Beijing, China: Association for Computational Linguistics, pp. 1681–1691. DOI: 10.3115/v1/P15-1162. URL: <https://aclanthology.org/P15-1162> (visited on 05/25/2022).

- Kiros, Ryan et al. (June 2015). “Skip-Thought Vectors”. In: *arXiv:1506.06726 [cs]*. arXiv: 1506.06726. URL: <http://arxiv.org/abs/1506.06726> (visited on 05/25/2022).
- Liu, Yinhan et al. (July 2019). “RoBERTa: A Robustly Optimized BERT Pretraining Approach”. In: *arXiv:1907.11692 [cs]*. arXiv: 1907.11692. URL: <http://arxiv.org/abs/1907.11692> (visited on 05/25/2022).
- Madhani, Nitin (Sept. 2007). “Getting started on natural language processing with Python”. In: *XRDS: Crossroads, The ACM Magazine for Students* 13.4, p. 5. ISSN: 1528-4972. DOI: 10.1145/1315325.1315330. URL: <https://doi.org/10.1145/1315325.1315330> (visited on 05/19/2022).
- Natural Language Processing(NLP)* (Oct. 2021). *Natural Language Processing (NLP): 7 Key Techniques*. en. URL: <https://monkeylearn.com/blog/natural-language-processing-techniques/> (visited on 05/25/2022).
- A Python Book* (June 2012). *A Python Book: Beginning Python, Advanced Python, and Python Exercises*. URL: https://web.archive.org/web/20120623165941/http://cutter.rexx.com/~dkuhlman/python_book_01.html (visited on 05/23/2022).
- Raffel, Colin et al. (July 2020). “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”. In: *arXiv:1910.10683 [cs, stat]*. arXiv: 1910.10683. URL: <http://arxiv.org/abs/1910.10683> (visited on 06/23/2022).
- Reimers, Nils and Iryna Gurevych (Aug. 2019). “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *arXiv:1908.10084 [cs]*. arXiv: 1908.10084. URL: <http://arxiv.org/abs/1908.10084> (visited on 05/25/2022).
- The Stanford Natural Language Processing Group* (2022). URL: <https://nlp.stanford.edu/projects/snli/> (visited on 05/25/2022).
- Vaswani, Ashish et al. (Dec. 2017). “Attention Is All You Need”. In: *arXiv:1706.03762 [cs]*. arXiv: 1706.03762. URL: <http://arxiv.org/abs/1706.03762> (visited on 05/23/2022).
- Yang, Yinfei et al. (Apr. 2018). “Learning Semantic Textual Similarity from Conversations”. In: *arXiv:1804.07754 [cs]*. arXiv: 1804.07754. URL: <http://arxiv.org/abs/1804.07754> (visited on 05/25/2022).

ÖZGEÇMİŞ

BURAK KESKİN

ÖĞRENİM BİLGİLERİ

Yüksek Lisans	Akdeniz Üniversitesi
2019-2022	Fen Bilimleri Enstitüsü, Bilgisayar Mühendisliği Bölümü, Antalya
Lisans	Türk Hava Kurumu Üniversitesi
2013-2018	Mühendislik Fakültesi, Mekatronik Mühendisliği Bölümü, Ankara

ESERLER

Ulusal bilimsel toplantılarda sunulan ve bildiri kitaplarında basılan bildiriler

- 1- B. Keskin and M. Gunay, "A Survey On Computerized Adaptive Testing," 2021 Innovations in Intelligent Systems and Applications Conference (ASYU), 2021, pp. 1-6, doi: 10.1109/ASYU52992.2021.9598952.