C. ÇAĞLAYAN

A NOVEL DEEP LEARNING APPROACH FOR CONTROLLED MULTI-TOPIC TEXT GENERATION

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

CANSEN ÇAĞLAYAN

A MASTER OF SCIENCE THESIS
IN
THE DEPARTMENT OF COMPUTER ENGINEERING

SEPTEMBER 2022

ATILIM UNIVERSITY 2022

A NOVEL DEEP LEARNING APPROACH FOR CONTROLLED MULTI-TOPIC TEXT GENERATION

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
ATILIM UNIVERSITY

BY

CANSEN ÇAĞLAYAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

SEPTEMBER 2022

Approval of the Graduate School of Natural and Applied Sciences, Atilim University.

Prof. Dr. Ender Keskinkılıç
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of **Master of Science in Computer Engineering Department, Atilim University**.

Assoc. Prof. Dr. Gökhan Şengül
Head of Department

This is to certify that we have read the thesis A NOVEL DEEP LEARNING APPROACH FOR CONTROLLED MULTI-TOPIC TEXT GENERATION submitted by CANSEN ÇAĞLAYAN and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Kasım Murat Karakaya
Supervisor

Examining Committee Members:

Assoc. Prof. Dr. Gökhan Şengül
Computer Engineering, Atilim University

Assoc. Prof. Dr. Kasım Murat Karakaya
Computer Engineering, Atilim University

Asst. Prof. Dr. Can Güldüren
Computer Technologies, Ufuk University

Date: September 14, 2022

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : CANSEN ÇAĞLAYAN

Signature :

ABSTRACT

A NOVEL DEEP LEARNING APPROACH FOR CONTROLLED MULTI-TOPIC TEXT GENERATION

Çağlayan, Cansen

M.S., Department of Computer Engineering

Supervisor : Assoc. Prof. Dr. Kasım Murat Karakaya

September 2022, 95 pages

One of the most important tasks in the Controllable Text Generation (CTG) domain is to create topic-controlled texts. In this study, we propose and design three different approaches, and conduct extensive experiments on them to observe the performance of the controlled multi-topic reviews generated in Turkish. In the first approach, we generate controlled multi-topic text using a single-layer GPT language model by incorporating several control techniques. To control the language model, we first add topic information to the sequential input, as a second technique we add the automatically extracted keywords for each topic to the sequential input in addition to the first technique. The last technique that we propose is a novel sampling strategy. We propose to use a topic selection classifier that enables the next token selection according to the probability of the selected tokens being on the desired topic. Then, we apply these approaches to a more advanced language model, the multi-layer GPT, and interpret the results. In addition to these experiments, we compare three different deep learning text classification models in order to create a reliable multi-topic review classifier.

Keywords: Controllable text generation, text generation, text classification, nlp.

ÖZ

KONTROLLÜ ÇOK KONULU METİN ÜRETİMİ İÇİN YENİ BİR DERİN ÖĞRENME YAKLAŞIMI

Çağlayan, Cansen

Yüksek Lisans, Bilgisayar Mühendisliği

Tez Yöneticisi : Doç. Dr. Kasım Murat Karakaya

Eylül 2022, 95 sayfa

Kontrollü Metin Üretimi alanındaki en önemli görevlerden biri konu kontrollü metinler yaratmaktır. Bu çalışmada, Türkçe olarak üretilen kontrollü çok konulu metinlerin performansını gözlemlemek için üç farklı yaklaşım öneriyor, tasarlıyor ve bunlar üzerinde kapsamlı deneyler yapıyoruz. İlk yaklaşımda, üç kontrol tekniğini birleştirerek tek katmanlı bir GPT dil modeli kullanarak kontrollü çok konulu metin üretiyoruz. Dil modelini kontrol etmek için önce sıralı girişe konu bilgisi ekliyoruz, ikinci teknik olarak ilk tekniğe ek olarak sıralı girişe her konu için otomatik olarak çıkarılan anahtar kelimeleri ekliyoruz. Sunduğumuz son teknik, yeni bir örnekleme stratejisidir. Seçilen belirteçlerin istenen konuda olma olasılığına göre bir sonraki belirteç seçimini sağlayan bir konu seçim sınıflandırıcısı kullanmayı öneriyoruz. Ardından, bu yaklaşımları daha gelişmiş bir dil modeli olan çok katmanlı GPT'ye uygulayıp ve sonuçları yorumluyoruz. Bu deneylere ek olarak, güvenilir bir, çok konulu metin sınıflandırıcısı oluşturmak için üç farklı derin öğrenme metin sınıflandırma modelini karşılaştırıyoruz.

Anahtar Kelimeler: Kontrollü metin üretimi, metin üretimi, metin sınıflandırma, nlp.

To my dear father. . .

ACKNOWLEDGMENTS

First of all, I would like to express my gratitude to my supervisor, **Assoc. Prof. Dr. Kasım Murat KARAKAYA**, for supporting and guiding me in every way during this process. I feel very lucky to have had the opportunity to benefit from his knowledge and experience. Throughout my life, I will be grateful to him for all the valuable lessons he gave me during this process.

I would also like to thank my beloved father **Mustafa ÇAĞLAYAN** who is the light of my life and my family for their unwavering support.

I shall also thank my dear friends **Mahmut YILMAZ**, **Meriç KALE**, **Ozan Can ACAR** and **Tolga ÜSTÜNKÖK** for always being there for me.

Furthermore, I thank the members of my thesis jury **Assoc. Prof. Dr. Gökhan ŞENGÜL**, **Assoc. Prof. Dr. Kasım Murat KARAKAYA**, and **Asst. Prof. Dr. Can GÜLDÜREN** for their time and suggestions.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

In this chapter, we summarize the work done in this thesis. After the text generation and controlled text generation tasks are explained in general, definitions of frequently used terms are given in the following sections. Then, the text corpus and evaluation metrics used in this thesis, which will be explained in detail later, are presented. After explaining the aim of the thesis and the contributions, the organization of the chapters of the thesis are provided.

1.1 Automatic Text Generation

In recent years, large text data have emerged all over the world due to the intense use of the internet. Processing these large text data, making them meaningful, and using them in various researches or applications has created a very important area. Known as Natural Language Processing (NLP), this field consists of computational methods that allow language to be represented and analyzed automatically. Automatic Text Generation (ATG) is one of the challenging tasks that need improvement in this area. Basically, the purpose of ATG is the ability of machines to generate meaningful texts using human languages. There are some tasks where ATG is applied. For example, Machine Translation, Text Summarization, Question Answering, Dialogue Systems (Chatbots), or Creative Writing (stories, poems, screenplays, etc.) are the most popular ones. Input data for ATG can be numerical, sound, picture, or graphic (non-linguistic input), but the most commonly used form is textual data. This situation can be referred to as a text-to-text generation. Accordingly, the model created in ATG takes the text, that is, a series of characters or words (sequences), as input,

transforms the input text into semantic expressions and learns to generate the output text. Machine Translation generates text in a different language based on the source text; Text Summarization creates a shortened version of the source text to include important information; Question-Answering (QA) generates textual answers to given questions; The dialog system enables chatbots to be used to communicate with people through generated responses [1]. ATG is one of the most open research areas in NLP to improve due to the increase in its usage in daily life. In ATG, there is a need for more progress every day in order to make the text generated realistic, meaningful and controlled.

Text generation studies are developed by researchers for various purposes, by applying different methods. Especially with the widespread use of Deep Learning methods and their application in this field, very promising results have been obtained. Recurrent Neural Network (RNN) based encoder-decoder models [2], Convolutional Neural Network - CNN based encoder-decoder models [3], Generative Adversarial Networks - GAN [4], Reinforcement Learning [5] and Transformer [6] are commonly used for this task. According to the most recent studies, Transformer based models have obtained very successful results in text generation, as in most NLP applications. Pretrained Language Models like GPT [7], BERT [8], etc. prepared using transformer-based and large text corpus are the most important examples of today. The model used in ATG is called the Language Model (LM). A LM, simply learns the probability of occurrences of the token, i.e. the desired split smallest structure (symbol, character or word) of a piece of text, based on given examples. The LM needs to learn the basic features of natural languages such as grammar, syntax, and semantics. This way, meaningful texts can be generated automatically. Before attempting to create controlled text, it is ideally expected that the LM be able to generate understandable, readable, linguistically appropriate text. Therefore, it is very important to understand and correctly construct the LM.

In this thesis, we first create a LM by using a single layer GPT transformer structure in its simplest form. Then, in order to use a more advanced language model, we train a multi-layered GPT architecture from scratch. The language models are explained in detail in the following sections.

1.2 Controlled Text Generation

Text generation that includes the desired features is called ”Controllable Text Generation”. Controllable Text Generation (CTG) aims to generate texts whose qualities can be controlled. The qualities to be controlled can be stylistic features such as politeness, sentiment, and formality; or maybe the characteristics of the hypothetical person writing the text, such as age, gender and character; or maybe the topic, keywords, and information in the content of the text [9]. In this thesis, topic controllable text generation is discussed. Accordingly, it is tried to ensure that the generated text by the LM is on the desired topic. This task is called controlled multi-topic text generation because more than two topics are being covered. In this sense, different techniques can be used to control the generated text. The techniques will be discussed in this thesis can be summarized as the modifications to be made with the input during the training of the language model and the changes to be made in the token selection (sampling) in the inference part after the training. With these techniques the controlled multi-topic text generation can be emphasized. In this thesis, we propose and compare three controlled multi-topic text generation techniques that we conducted extensive experiments on.

1.3 Definition of Terms

As in most NLP applications, some frequently used terms and their explanations on text generation are as follows:

- Sequence : A series of characters, words or sentences.
- Token : Smallest structure (symbol, character or word) of a piece of text.
- Corpus : Language resource consisting of a large and structured set of texts.
- Language Model : Model that learns the probability of occurrences of the token based on given corpus.
- Prompt : Inital text input to the language model.
- Tokenization : Seperating text into tokens.

- Vocabulary : Maximum n number of tokens in the corpus.
- Text Vectorization : Turning text to numeric representations that can read by an embedding, dense layers etc.
- Sampling : Selecting the next token from probability distribution. It can be done in various ways. For example; Greedy Search (Maximization), Temperature Sampling, Top-K Sampling, Top-P Sampling (Nucleus sampling), Beam Search etc.

1.4 Aim

The aim of this thesis is to generate controlled multi-topic text in Turkish by developing different techniques and language models. First, the experiments aims to generate text using a single-layer causal GPT transformer language model that we put all modules of GPT block. We apply three controlled text generation techniques on this language model. These techniques are modifying the sequential input with topic, modifying the sequential input with keywords, and modifying the sampling with topic selection classifier.

The first technique we use for controlled multi-topic text generation is to modify the sequential input, that is, to add the topic information to the input. Then, in order to improve this technique, we extract five keywords from each topic using TF-IDF (term frequency-inverse document frequency) [10] method and add them to the sequential input. Then, in addition to these two current techniques, we propose a new sampling strategy while creating the output text. For this, instead of making a random token selection from the highest probabilities, we create a structure it is ensuring that the token to be selected is in the desired topic with the help of a topic selection classifier. We add each technique to the previous one to get the generated text closer to the desired topic. Then we adopt another language model, train multi-layer causal GPT LM as a more advanced LM to increase the meaningful text generation capacity. We also apply, the sampling method that we propose to this model to generate controlled multi-topic texts.

In addition to these, we aim to develop a reliable text classification model in order to

measure whether the texts we generate are on the desired topic. Therefore we also conduct extensive experiments in text classification.

1.5 Text Corpus and Evaluation Metrics

We use the dataset containing Turkish reviews on 32 different topics called TC32 [11] in this thesis. The data analysis and multi-class text classification experiments are explained over the TC32 dataset. For text generation and controlled text generation experiments we create a sub-dataset that has 5 topics, called TC5. While creating the TC5, we select the most unrelated topics and eliminate the similar topics. For example, the topic of shopping makes it difficult to distinguish between topics as it suppresses many topics. In this way, we increase the training speed and topic sensibility of the language models.

Since we aim to generate controlled multi-topic texts, we need to measure whether the generated texts are on the desired topic. For this we use a reliable topic classification model. The purpose of this model is to determine the topic of the given text. Therefore, we compare Custom Transformer Encoder, Convolutional Neural Network (CNN), and Fine-tuned Turkish BERT (BERTurk) classification models for the multi-topic text classification task. Then, we examine the classification accuracy of the generated texts using the chosen classifier. Experiments for multi-topic classification model selection and the purpose of this task are explained in detail in Chapter 4.

Although the aim is to generate controlled multi-topic text in this thesis, it is also important to measure the quality of the generated texts. Some evaluations metrics used for this purpose are the BLEU score [12] and BERTScore [13]. BLEU is a precision-focused metric that simply counts the n-gram overlap of the reference and generated texts, ignores the meaning. On the other hand BERTScore computes the similarity of two sentences as a sum of cosine similarities between their tokens' embeddings [13]. However, both evaluation metrics can be more reliable when there are correct texts that can be referenced, basically as in tasks such as machine translation or caption generation. Therefore, the most important evaluation in the studies conducted in this thesis is human judgment.

1.6 Contributions

According to the literature research, we could not locate a study on Topic-Controlled Creative Text Generation task by using deep learning models in the Turkish language. To the best of out knowledge mostly there are studies that provide text generation in the fields of Text Summarization and Dialogue Systems (Chatbot). Since they do not serve exactly the same purpose, they have been excluded from our research area. In this case, our conference paper titled Topic-Controlled Text Generation, published at the 6th International Computer Science and Engineering Conference in 2021 [14] can be cited as an example of a study conducted in the field we examined and in Turkish. In this conference paper, the dataset used in this thesis which is Multi-Class Classification data for Turkish (TC32) [11] is also used and the first controlled multi-topic text generation technique that will be described here. In that study, we aim to create Turkish texts on the desired topic by using a single layer GPT transformer language model known as Miniature GPT [15]. The controlled multi-topic text generation technique used for the paper is the updating the sequential input technique, which is one of the most basic approaches in the field of controllable text generation. According to this technique, we prepare the topic information, add it to the text input representation at each step of training, and give the combined version of the two to the single GPT block. Although we are not able to quantitatively measure the results in our paper we observe that, our model not only generates texts containing the characteristics of the Turkish language but also uses a few words that evoked the desired topic. Despite the success of this first step we have taken, there has been the generation of many meaningless texts that do not evoke the desired topic. One of the reasons for this is the need to try different techniques in order to increase the performance in addition to this most basic approach we use in controlled text generation. This technique will be explained in the Chapter 6.

As a second controlling technique, we develop the current model by extracting the keywords of topics and adding them to the inputs. In this way, although there is an improvement in the model, we achieve the real success with the addition of the third technique. The third technique is a new sampling approach for controlled multi-topic text generation which is called *Sampling with Topic Selection Classifier*. With this

technique is we achieve the highest success according to a text classifier that finds whether the generated text belongs to the desired topic or not. In this technique we make some changes after the *Top-k sampling*, that is, the selection way of the token to be generated. As far as we know, this is a new approach to ensure that the token to be selected after top-k is in the desired topic with the help of a topic selection classifier. After the experiments of this language model and technique, we apply propose technique to more advanced model than the first one.

1.7 Organization of the Thesis

This thesis is organized as follows. Chapter 1 introduces the general subjects, aim, and contributions of the study. Chapter 2 gives a summary of the previous work done on text classification, text generation, controllable text generation, and topic-controlled text generation. Chapter 3 describes data analysis of the dataset. Chapter 4 summarizes the experiments and studies carried out to determine the multi-topic text classification with three different classification models, which will help to understand whether the generated texts are on the desired topic. Chapter 5 presents the first language model used for text generation which is a single layer GPT transformer structure without control. Also explains the text generation task and evaluation of generated texts. Chapter 6, at the first part, provides the generation of controlled multi-topic text generation, using the first language model with modifications to be made in the sequential input and the sampling method. Second part of Chapter 6, describes using more advanced language model and applying the controlled multi-topic text generation technique which we propose. Finally, Chapter 7 explains the results and concludes the thesis by summarizing the study.

CHAPTER 2

LITERATURE REVIEW

In this section, we give a brief analysis of the literature on the subjects that are covered in general. In the thesis, firstly, we explain the experiments on the text classification task, so this section will also start with text classification. Later, continues with text generation and controllable text generation tasks. Since we explain these subjects in detail in the following chapters, examples from the literature are given here.

2.1 Text Classification

Text Classification is a semi-supervised machine learning task that automatically assigns a given document to a set of predefined categories based on text content and extracted features [16]. Different methods have been used so far for text classification. The word matching method is the first proposed classification algorithm. While this method determines whether a document belongs to a category, it only looks for identical or similar words from the same class in the document. Until the 1990s, text classification methods based on statistics and machine learning gradually emerged, in this way, some rules can be deduced from the document and classified, and then the classifier can be trained according to those rules to take its final form [17]. However, it may take human work to get some rules out. Today thanks to the of deep learning, much more effective, fast results can be obtained. Deep learning techniques are trained using large sets of labeled data and neural network framework learn the features right from the data without the requirement for manual feature extraction methods [18]. Figure 2.1 shows the basic architecture of deep learning-based approach for text classification task. Recurrent Neural Network (RNN), Convolutional Neural

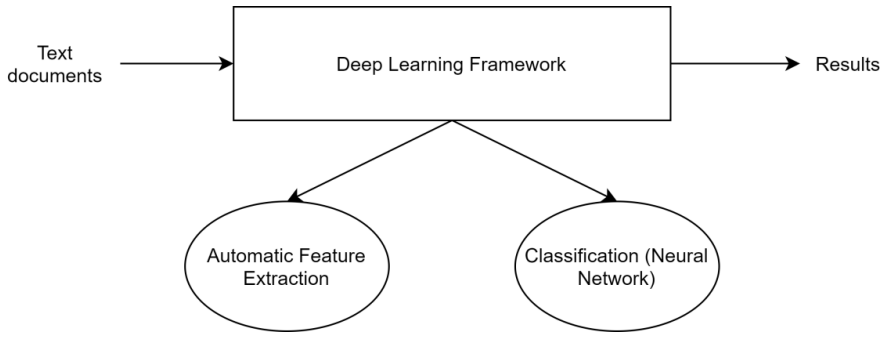

Figure 2.1: Basic architecture of deep learning-based text classification [18].

Network (CNN) based deep learning models or combinations are generally used in this regard. Thanks to the impact transformers [6] brought to the NLP world in 2017, the most common models for text classification today are transformer-based.

As explained in detail in Chapter 4, our purpose of using text classification in this thesis is that we need a structure that will automatically find whether the texts we generate as controlled multi-topic are in the desired topic. Using this text classification model, we can measure the impact of our controlled text generation techniques on the generated text. Accordingly, we develop and compare the three most commonly used models in the text classification task. These models are Convolution Neural Network (CNN), Custom Transformer Encoder (CTE) and pre-trained Turkish BERT (BERTurk) model.

CNN is first proposed to apply for character recognition [19] for NLP tasks. Then it is proposed for the text classification and [20] with using a single-layer convolution neural network in many classification datasets and achieved successful results. CNN involves a series of filters of different sizes and shapes which convolve (roll over) the original sentence matrix to reduce it into further low dimension matrices and in text classification CNN are being applied to distributed and discrete word embedding [21]. In this way, it can get successful results in the text classification tasks [22] [23] [24]. CNN is also used by for putting labels to the semantic role [25] and as a encoding technique for the text classification [26].

In this thesis, except for CNN, the models use for text classification task are transformer-based models. Transformer is basically an encoder-decoder model consisting of multi-

head self-attention and fully connected feed-forward network structures. Although originally offered for machine translation, it is used today in many different NLP tasks. The encoder structure of the transformer model can be used in the task of text classification since it will extract the representation of the words in the inputs by taking into account the relationship between the words in the inputs with self-attention mechanism. Therefore, we use the Custom Transformer Encoder (CTE) model, which consists of a single layer transformer encoder block. The last model we use is the BERTurk [27] [28], which is a pre-trained BERT (Bidirectional Encoder Representations from Transformers) [8] model trained in Turkish. BERT is a transformer encoder based model which can generate contextualized word vectors and it is a significant turning point in the development of text classification task. It has been studied by many researchers [29] [30] and achieves better performance than the most classification models [31].

In this thesis, we consider the studies done in Turkish, as we classify the Turkish controlled multi-topic texts. In some studies on text classification in Turkish, we observe the performance of transformer-based models. For example, in [32] various models has been tried to detect fake news in Turkish language. In this study machine learning algorithms (i.e. Naive Bayes, Random Forest (RF), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression), deep learning algorithms (i.e. Long Short-Term Memory (LSTM), Bi-directional LSTM, CNN, Gated Recurrent Unit (GRU) and Bi-directional GRU) and deep learning based language transformers (i.e. BERT and its variations) are compared to the most frequently used evaluation metrics in machine learning literature for classification problems (*Accuracy*, *F1 score*, *Precision* and *Recall*). The transformer-based BERTurk model give the best results. In [33] , topic classification is made using Turkish tweets, traditional machine learning methods (SVM, Naive Bayes and Random Forest) and custom transformer encoder model. TCE shows the best performance compared to classification metrics. In, [34] text classification is made over the book explanations in Turkish and the categories they belong to. For this task, LSTM, CNN and TCE models are compared and the TCE gives the most successful results according to classification metrics. Also there are many studies conducted using the BERTurk model [35] [36] [37] [38] [39] [40].

2.2 Text Generation

Text generation is generally the task of automatically generating meaningful output texts from a given input text. With the recent resurgence of deep learning, deep neural automatic text generation NLG models have achieved remarkable performance in enabling machines to understand and generate natural language [1]. There is a lot of research in this field, especially in English, using various methods and models [41] [42] [43] [44].

Today, the most successful ATG operations are performed with transformer-based Pre-trained Language Models (PLMs). PLMs aim to learn universal language representation by conducting self-supervised training on large-scale unlabeled corpora [1]. Pretrained on the large-scale corpus, PLMs are able to understand natural language accurately and express the human language fluently, both of which are critical abilities to fulfill the text generation tasks [45]. PLMs are known to be successful in various text generation sub-tasks [46] [47] [48].

One of the most advanced pre-trained neural language model is GPT [49]. GPT is a transformer decoder based pre-trained model that can be used ATG. GPT can generate long texts that are almost indistinguishable from human-generated texts [50]. Empathetic social chatbots, such as XiaoIce [51], seem to understand human dialog well and can generate interpersonal responses to establish long-term emotional connections with users [52]. Considering the success of GPT models in text generation, we use GPT language model (LM) structure in this thesis to generate text in Turkish. For this, first, we prepare Miniature GPT which consists a single-layer GPT block LM with each module accessible, and generated uncontrolled text, and then we apply the three controlling techniques on this model. Then, in order to increase the quality of the generated texts, we train the multi-layer original GPT architecture from scratch.

2.3 Controllable Text Generation

Controllable Text Generation (CTG) is the task of generating natural sentences whose attributes can be controlled [9]. In general, the properties of the generated texts are

not controlled, so CTG is an important research area today. Many studies are devoted to controlling the content of the generated text [53] [54] [55]. The control elements provided on the texts generated can be things like stylistic [56], the semantics [57] or sentiment [58].

In this thesis, we consider the topic controllable text generation. Accordingly, the automatically generated text should be on the desired topic. Topic-controlled text generation can be done using a variety of models and techniques [59] [58] [60] [61].

In this thesis, we first use the modifying the sequential input technique. This technique is basically adding a control factor to the embedding representations of the inputs. We apply this technique by first concatenating the topic information with each sequential input at each time step and then concatenate the keywords that we automatically extract for each topic in addition to the topic information.

Noraset et al. [62], use this technique for the task of definition modeling, they concatenate word embedding vectors of the word to be defined, Zhou et al. [63] concatenate the hidden representation of the external source of information for the dialogue response generation, Prabhumoye et al. [64] concatenate the hidden representation of the external source of information to Wikipedia update for the generation process, Harrison et al. [65] concatenate style and personality constraint into the generation process, Chandu et al. [66] concatenate the personality representation for the story generation process [9].

There are some examples in Turkish text generation [67] [68] [69] [70] [71] [77]. The summarization task is dominant for this domain in Turkish [72] [73] [74] [75] [76]. In addition, as far as we know, there are very few published research on controllable text generation in Turkish.

CHAPTER 3

DATASETS

We use Multi-Class Classification data for Turkish (TC32) [11] as a dataset in every experiment in this thesis. It is a benchmark dataset for Turkish text classification task. The reason why we chose a Turkish dataset is that we have not come across a topic-controlled creative text generation study in Turkish. Although there are successful studies on this subject in English, we want to discover how it would yield results in Turkish. In the controlled multi-topic text generation, which is the main purpose of this thesis, we use this dataset of multi-topic Turkish reviews to test both the existing controlling techniques and the technique we propose.

TC32, contains $430k$ reviews for a total of 32 topics (categories). There are about $13k$ reviews on every topic. It is quite sufficient to use it for controlled multi-topic text generation and multi-topic text classification tasks. It is very important to understand the data very well in every problem where machine learning or deep learning studies will be done. The success of the models depends on the understanding and correct pre-processing of the data. In addition, values such as the maximum sequence length for the input to be used in the training of the models for both text generation and text classification tasks are decided in this way. Therefore, the data needs to be examined and analyzed in detail.

There are three main NLP tasks covered in this thesis. The first of these is text generation task to generate Turkish texts thanks to a language model developed without any control, the second one is controlled multi-topic text generation task, which ensures that the generated text is on the desired topic, and the last one is the text classification task that allows us to understand whether the texts generated under topic control are

really on the desired topic. Data preparation processes required for these three tasks is differ. We use the original 32 topics dataset, TC32 for the text classification but we create a sub-dataset from the original one for the controlled text generation because some classes are semantically very dominant in the TC32, they reduce the sensitivity of the language models used in controlled text generation. We select 5 unrelated topics (finance, health, tourism, sport, food) for this sub-dataset and it is called TC5. TC5 will be explained in the text generation part but the TC32 will be reviewed in general in this chapter.

3.1 Topics and Reviews

TC32 consists of two columns, category (topic) and text (review). Examples from TC32 :

- category : *alisveris*

  text : *"Boyner Siparişimi İptal Edemiyorum,1007007428 numaralı 08.06.2029 tarihli siparişimde 3 al 2 ode kampanyasında ürünlerimi alırken birisi sepetten çıkmış ve fiyat fark etmediği için anlamadım ve ödedikten sonra baktım 2 ürün vardı sepetimde. İptal etmek istiyorum ama müşteri hizmetleri çalışmıyor. Mail attım dönüş yok. İkinci kez yazd...Devamını oku"*

- category : turizm

  text : *"ETS Tur İptal Talebi,Eşimle 14.03.2020 - 15.03.2020 tarihli Ankara Princess Otel'den rezervasyon yaptırdık. İkimizden de para kesintisi oldu adımıza 2 oda ayrıldı. 1 Odayı iptal ettirmek istediğimizi ETS'ye bildirmemize rağmen olumsuz geri dönüş aldık. Otelden iptalini sağladık ama ETS gereken özeni göstermiyor.Devamını oku"*

- category : *ulasim*

  text : *"Anadolu Jet Bilet Değişikliği Yapamıyor!,Ucuza biletten anadolu jet uçağı İzmir İstanbul seferine bilet aldım. Bilet 156. Tl'lik. Ben biletimi açığa aldırmak istiyorum sadece ve bana 115 TL kesinti olur diyorlar şaka gibi bilet*

zaten 156 TL. İptalde etmiyorlar iade sağlamıyorlar. Anadolu Jet'ten yapmaya çalıştığında acente bileti yapamayız...Devamını oku"

- category : *mutfak-arac-gerec*

  text : *"Karaca Züccaciye Tost Makinasının Parası Ödendi Ürün Yok!,""Tost makinesi sipariş verdim. Kargo elemanı ürünü demir parmaklıktan aşağıya attı, kendisine elektrikli aleti nasıl atarsın dediğimde aldı geri ve o günden bugüne hiçbir bilgi yok, ürünün parası ödendi, ürünle ilgili hiçbir bilgi yok, muhtemelen parçalandı, Karaca Züccaciye teslim edildi diyor, edil...Devamını oku""*

- category : *gida*

  text : *"Nutella Puan Uygulaması,Nutella uygulamasından puanla sipariş oluşturdum kargo takip numarası yanlış Aralık 26 sında verdiğim sipariş hala kargolanmamış iki ürünümde yok ortada bu nasıl iş anlamıyorum boş yere mi o kadar uygulamayı indiriyoruz insanları kandırmak için mi çok ayıp.Devamını oku"*

The topics of the TC32 dataset and the number of reviews in each topic are shown in Table 3.1.

There are a total of 431306 reviews (texts) on 32 different topics in the TC32, but there are 427231 reviews uniquely and there are no null values. So this means that there are some repeating reviews. We remove these duplicate reviews so that a total of 427231 different reviews remain in the dataset. There are about $13k$ reviews on each topic. As can be seen from the visualization in Figure 3.1, it can be said that this dataset is balanced by looking at the number of reviews on each topic, the number of samples is almost evenly distributed over the topics. In this way, no class balancing is required for training of text generation, controlled text generation or text classification models.

First of all, to analyze the reviews kept as text, we check how many words are in each review. Figure 3.2 shows the number of words of the random 10 reviews from TC32 in the words column. We observe and analyze the distribution of the number of words in all reviews. These analyzes directly affect the parameter selections in the operations to make on the text data.

If we look at the review lengths in terms of the analysis of words in reviews, as seen in Table 3.2, there are 183 words in the longest review and 2 words in the shortest

Table 3.1: Number of reviews for each topic for 32 topics dataset.

Topic	Number of Reviews
kamu-hizmetleri	13998
cep-telefon-kategori	13975
enerji	13968
finans	13958
ulasim	13943
medya	13908
kargo-nakliyat	13877
mutfak-arac-gerec	13867
alisveris	13816
mekan-ve-eglence	13807
elektronik	13770
beyaz-esya	13761
kucuk-ev-aletleri	13732
giyim	13676
internet	13657
icecek	13564
saglik	13559
sigortacilik	13486
spor	13448
mobilya-ev-tekstili	13434
otomotiv	13377
turizm	13317
egitim	13264
gida	13150
temizlik	13111
mucevher-saat-gozluk	12964
bilgisayar	12963
kisisel-bakim-ve-kozmetik	12657
anne-bebek	12381
emlak-ve-insaat	12024
hizmet-sektoru	11463
etkinlik-ve-organizasyon	11356

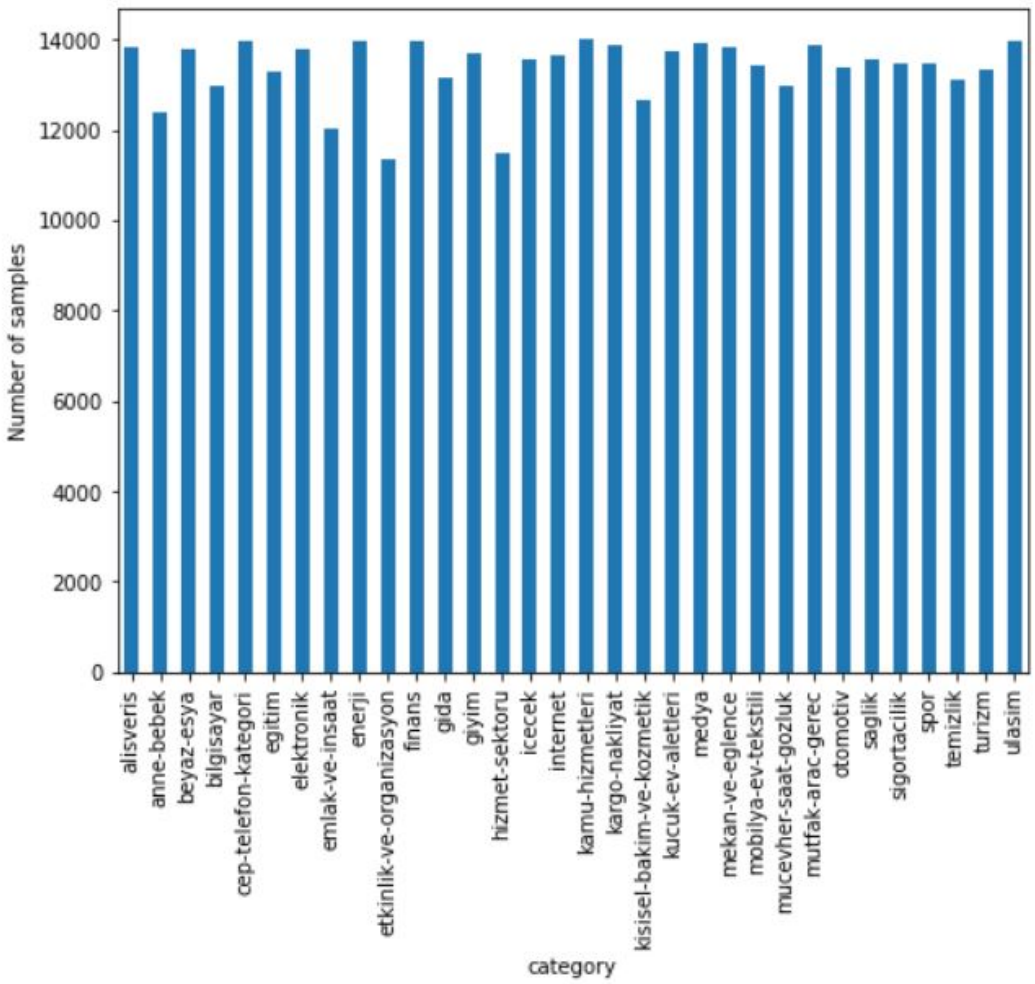

Figure 3.1: Distribution of reviews for each topic.

category	text	words
sigortacilik	Halk Sigorta Sigorta İşlemleri,Bundan önceki aracımı 5. Basamaktan sigorta yaptırırken şimdi onu satıp başka araç aldım sigortam 4. basamaktan işlem görüyormuş sigorta şirketlerini anlamış değilim gereğini bilgilerinize sunarım ilginiz için teşekkür ederim.Devamını oku	33
elektronik	Philips TV 49PUS6581 Ekranda Siyah Leke Sorunu,"Televizyonumun hemen hemen her köşesinde ve de kenarlara yakın yüzeylerde siyah noktalar oluştu. Televizyonu alalı 2,5 sene oldu. Garantisi dolar dolmaz bu sorunla karşılaştım. Özellikle beyaz renkli ekran ve maçlarda çok can sıkıyor. Lekeler gittikçe artıyor. Çözüm bulunmasını rica ederim.Devamını oku"	47
alisveris	Media Markt Kayıp Play Station,"Merhaba ps4 satın aldım 1 hafta geçmesine rağmen ulaşmadı. Bunun dışında kargo takip numarası belirleyemedikleri için ürünün durumu belli değil. Yakışmadı böyle markaya, mağazalarında kaybettiği imaj online da yerle bir oldu. Çok başarısızlar çokDevamını oku"	39
anne-bebek	Joker Mağazası'na Verdiğim Sipariş Gelmedi,90 lı can bebe 54.95 yazıyordu 2 paket sipariş verdim 2 paket 90 lı can bebe geleceğine 2 paket 76 lı geldi sipariş verirken peşin ödeme tek çekim almak güzelde ürünü internete verdiğiniz gibi göndermek mi zor geliyor. Madem 76 lı göndereceksiniz 90 lı can bebe niye yazıyor özelliklerinde baştan aşa...Devamını oku	56
cep-telefon-kategori	Huawei Parmak İzi Okumuyor, Hata Veriyor,"Huawei Y7 2018 model telefonum parmak izi okuyucu yeri çalışmıyor, sürekli hata veriyor. Parmağımı koyduğumda parmak izi donanım kullanılmıyor yazıyor. Huawei'den hicare'de canlı sohbet yaptım hanımefendi telefonun sistem ayarlarını sıfırlamamı istedi, fakat sonuç yine aynı parmak izi okumuyor.Devamını oku"	44

Figure 3.2: Number of words for random 10 reviews.

review. Also, 75% of reviews have less than 50 words. Figure 3.5 shows graphical representation of this situation. As we discuss in the next sections, based on this information, during the training of text generation and text classification models, we

Table 3.2: Analysis of words in reviews.

mean	44.408624
std	8.108499
min	2.000000
25%	42.000000
50%	46.000000
75%	49.000000
max	183.000000

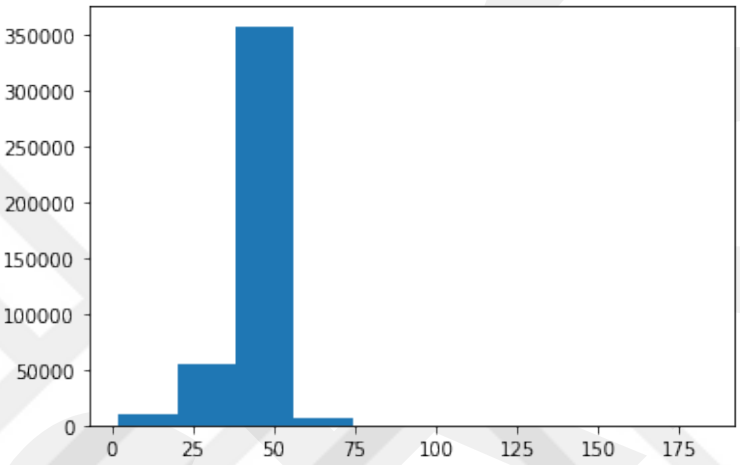

Figure 3.3: Distribution of number of words in reviews.

adjust each input sequence to contain 40 words because Figure 3.3 shows that the majority of reviews are about 50 words or less.

Very short and very long texts in NLP tasks may need to be analyzed and removed sometimes. Reviews shorter than 15 words according to the threshold value determined for TC32 are shown in Figure 3.4. These short reviews may be omitted from the dataset in some cases like text classification task on the grounds that they do not contain enough information. However, the studies conducted in this thesis, short reviews are not excluded from the dataset, they are padded. Also the use of each text in the text generation task is important for the learning capacity of the language model. Only during the text vectorization process, we trim very long texts to the desired maximum number of words which is 40.

category	text	words
alisveris	Vatan Bilgisayar Kargo Hakkında Bilgi Verilmemesi Çağrı Merkezinin 30 Dakika Bekletmesi,"02.05.2020 tarihinde verdiğim	13
alisveris	A101 Sipariş Alıyor Ancak Getirmiyor,"28.05.2020 Tarihinde sipariş verdim.	8
alisveris	Boyner Eksik Ürün Gönderme,"Merhabalar	4
alisveris	Tekzen 40 Gündür Ücret İadesi Yapılmıyor,"17.04.2020 tarihinde jaluzi perde siparişi verdim. Sipariş numaram: 636576660	14
alisveris	Boyner Sipariş 1 Haftadır Tedarik Sürecinde!,"Merhabalar,	6
...	...	...
ulasim	Pegasus Uçuşu İptal Etti!,"13.02.2020 diyarbakır'dan istanbul sabiha gökcen havalimanına 21.25 uçuşumuz vardı.	12
ulasim	Kent Kart'a FastPay İle Yaptığım Yükleme Hesabıma Geçmedi!,"13/02/2020 Tarihinde Denizbank'ın FastPay uygulaması vasıtasıyla	13
ulasim	Efe Tur İnternet Yok,"İnternet şifresi istiyoruz kota doluyor veremeyiz diyorlar televizyonlar açılmıyor	12
ulasim	Efe Tur'da Saygısızlık, İhmalkarlık!,"11.02.2020 20:00 Ankara İstanbul seferini yapan 210 numaralı aracın şoförü	13
ulasim	Pegasus Koltuk Numarası Değişimi ve Kırık Valiz,"	7

Figure 3.4: Some short reviews.

3.2 Vocabulary

In this section, some analyzes will be explained for the words in the reviews, and then the data cleaning processes to be carried out in line with these analyzes are decided. Firstly, we observe that there are 900327 distinct words in the TC32. The main reasons for the emergence of so many distinct words are spelling mistakes, punctuation errors, counting numbers to words, and the perception of some Turkish words as different words because they are written using the English alphabet. For example, some unique words are: 'bağlamadı,satmış', 'yapılabile...devamını', 'olmuyor,telefonum', 'şebekelerimiz', 'oturuyoruz.!', 'kullanmazsan', 'değil,uraw', 'önemsenmeyen', 'gidersindevamını', 'sorunu,06.04.2020', 'xyz,', 'sütüme', 'odaklanamıyor', 'etmiyorlar,almış', 'bence', 'yollayacaksınız.devamını', 'libre', "gamepower'da", 'üretilmiştir...', '675', 'arayın,bu'. As can be seen, words combined with punctuation marks, numbers or misspelled are counted as a unique word. In order to correct this situation, it is important to perform data cleaning operations.

The 20 most frequent words in the data set are shown in Table 3.3. According to this, the most common word in the TC32 is seen as the word "oku" (read) in Turkish. This is because, the "Devamını oku"(Read more) statement is included at the end of each review in the raw TC32. This statement is an expression that remains to see the rest of each review while reviews are collected from various websites.

In addition, Turkish stop words and conjunctions can be seen in the most common

Table 3.3: Most frequently used 20 words in the TC32.

Word	Count
oku	382925
bir	290203
ve	254437
için	134568
bu	123079
de	111211
da	88801
ama	70306
yok	67821
ile	67620
gün	67342
aldım	65740
sonra	64183
ne	62723
rağmen	61891
TL	59720
önce	58567
tarihinde	57490
çok	56586
kadar	55290

words. Based on this analysis, we decide, it can be deduced that the phrase ”Devamını oku” and Turkish stop words should be cleared from the text for future tasks. As a result of the analysis, we use the TC32 dataset for text classification as stated before, while we divide a sub-dataset consisting of 5 topics (TC5) for controlled text generation tasks. Also, we perform data pre-processing separately for text classification and text generation tasks as a result of the analyzes and inferences we make here. These are described in Chapter 4 and Chapter 5.

CHAPTER 4

MULTI-TOPIC TEXT CLASSIFICATION

This chapter explains our experiments to obtain a reliable review classifier model. The reason for this task is to understand whether the generated reviews are on the desired topic when we generate controlled multi-topic reviews. In Chapter 6, we evaluate the topics of the reviews we generate using various models and techniques, thanks to the choosing review classifier experiments described here. In this section, after giving general information about the text classification task, we explain the purpose of this task in this thesis in detail and the experimental design. Then we share the data preparation, the models, and the results.

4.1 Text Classification Task

Text classification is one of the most basic and popular fields of study in Natural Language Processing (NLP). Text classification also known as Text Tagging or Text Categorization is the determination of the group or category to which the textual data (sentence, paragraph, document etc.) belongs. In recent years, with the development of the internet, there has been a great increase in textual data. Text data is created and used in many areas such as some websites used for various purposes like shopping, education, tourism, mail applications, social media platforms, emails, etc. Over time, the processing and classification of this huge text data has become a very important issue. Inferences can be made from these data for research or commercial purposes, and people's lives can be made easier in a wide variety of fields. Handling so much text data manually is quite troublesome. For this reason, more advanced systems are needed every day to process and classify the ever-increasing text data. Thus,

automatic text classification has taken its place as one of the most important research topics in the NLP. Automatic classification of texts is used in many areas in daily life and makes our life easier. The most well-known examples are sentiment analysis, topic classification, email filtering, language detection, news categorization etc.

Text classification is the task of classifying text documents under a predefined category. Basically, the text classification consists data pre-processing, feature Extraction, training the model and deploying the model steps. These steps also shown in the Figure 4.1. The pre-processing and feature extraction steps are very important for text classification task. The main purpose of pre-processing step is to examine the words in a sentence or document and simplify, normalize, and condense them for the training. Cleaning process for the text data (removing stopwords or wrong and unnecessary words, lowercase conversions, removing unnecessary characters such as punctuation and special characters, removing numbers) is required during the preparation process. Also, tokenization is a very important step. Tokenization is a preprocessing method which breaks a stream of text into words, phrases, symbols, or other elements called tokens [78]. String tokens cannot be directly read by machine learning models. Instead, they must be converted into a vector representation of these inputs [79]. In this way, text inputs are made to be understood and processed by a classification model by taking a numeric value. After that, formal feature extraction methods can be applied. The common techniques of feature extractions are Term Frequency-Inverse Document Frequency (TF-IDF), Term Frequency (TF) [10], Word2Vec [80], and Global Vectors for Word Representation (GloVe) [81][82]. Instead of these methods, an embedding layer can be trained to achieve word representations as we do in this study.

For the text classification, training can be performed after the text data preparation is complete. In supervised learning, the machine is presented with training examples consisting of input/output pair patterns where it is required to predict the output values of new examples based on their input values [84]. The input to the classifier is cleaned and vectorized text, each labeled with its own class label. The purpose of the model is to generate a description for each class in terms of various attributes. The trained model is then used to classify future records whose classes are unknown. There are many options for classification model selection. Although machine learning methods like Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, Random Forest, Sup-

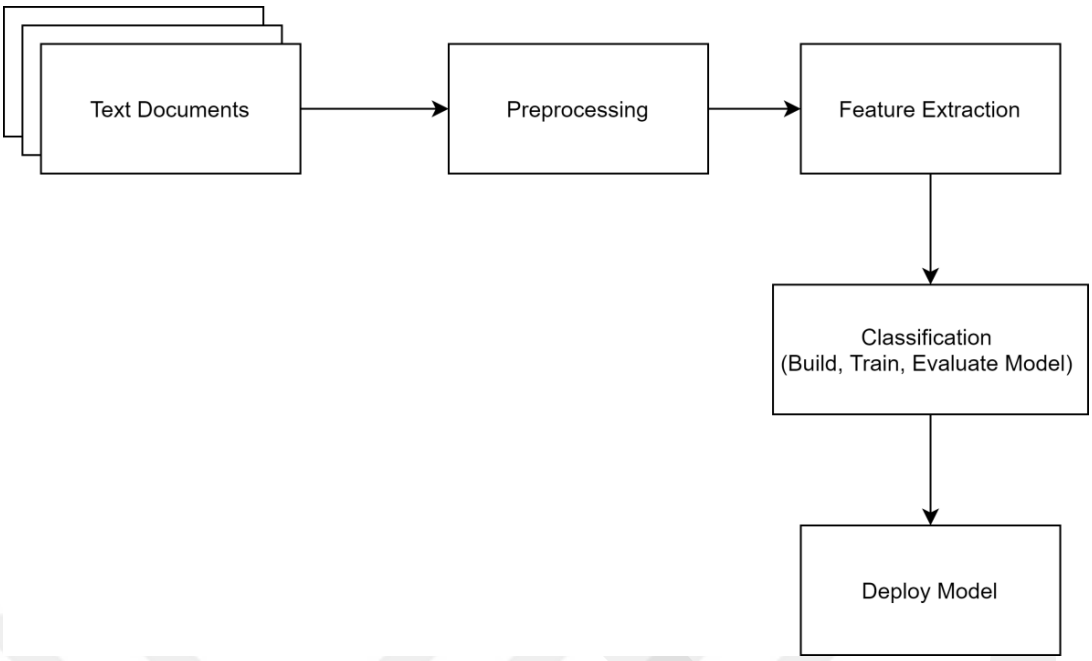

Figure 4.1: Basic architecture of text classification.

port Vector Machine are still used, they are a bit outdated and sometimes require high hardware power to handle multi-dimensionality. The most frequently used models that give fast, effective and successful results today are models using deep learning methods. Recurrent Neural Networks (RNN) and Convolutional Neural Networks (CNN) based models are frequently used and successful results are obtained. However, since the day it was introduced, state-of-the-art Transformer-based models, have had an important place in NLP. Of course, the same is true for text classification tasks. Therefore in this thesis, we make experiments with the Custom Transformer Encoder classification model, the One Dimensional Convolutional Neural Network (CNN) classification model and the Turkish BERT classification model, which is a pre-trained transformer-based classification model for text classification task and the results are compared. The results obtain using the TC32 dataset will be explained in detail in the following sections.

Text classification, like other classification tasks, is handled in three categories. These are binary classification, multi-class classification and multi-label classification. In binary classification, texts belong to one of two classes. For example, the spam-not spam email filter or sentiment analysis, in which the text is grouped as positive or

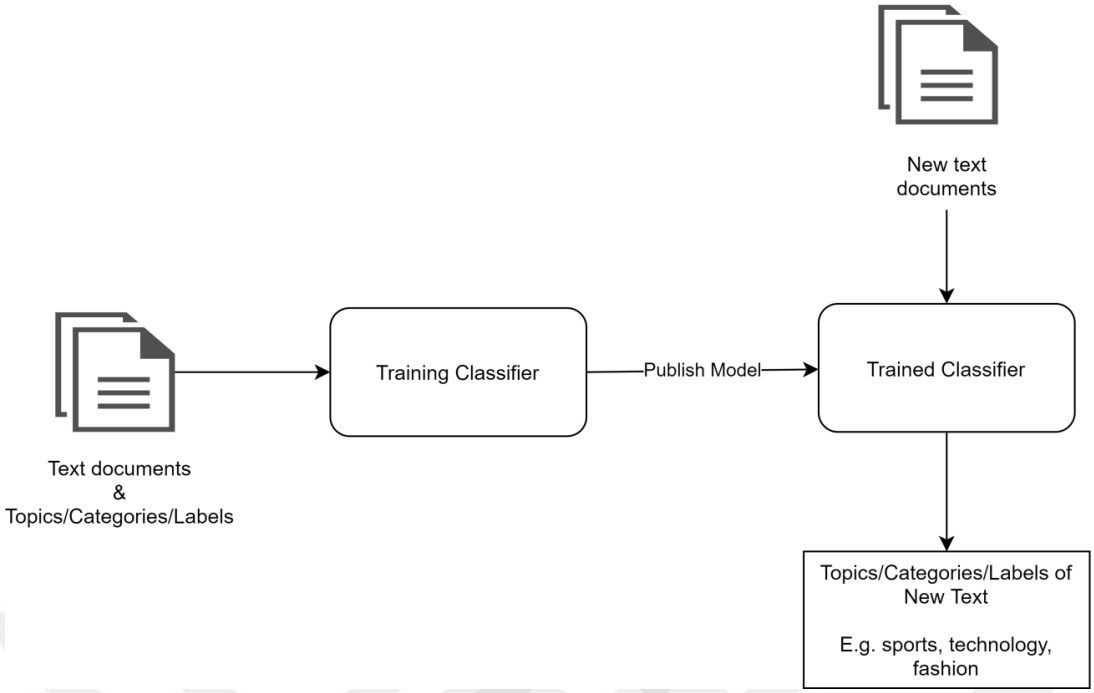

Figure 4.2: Basic architecture of multi-topic text classification task.

negative, can be given. In multi-class classification, there are more than two classes, however text can belong to only one of them e.g. some topic classification tasks. Multi-class classification is also known as a single-label problem. On the other hand, in multi-label classification, text can belong to more than one class at the same time. In this thesis, there are 32 classes and each data belongs to only one topic. Therefore it is a multi-class classification problem. When it is about classifying topic of the texts, it is called topic classification. It means, after the classifier is trained and finalized with the training set, it is expected that the model will be able to distinguish text data that it has not seen before and find out it's topic. TC32 dataset contains reviews on 32 different topics. In this way, it is a very suitable dataset for us to perform the topic classification task. The next chapters will describe multi-topic classification operations with TC32 dataset. The basic structure of multi-topic classification is shown in Figure 4.2.

4.2 Purpose and Experiment Design

The purpose of using the multi-topic classifier in this thesis is to measure whether the generated text on the desired topic really belongs to that topic. Thanks to the multi-

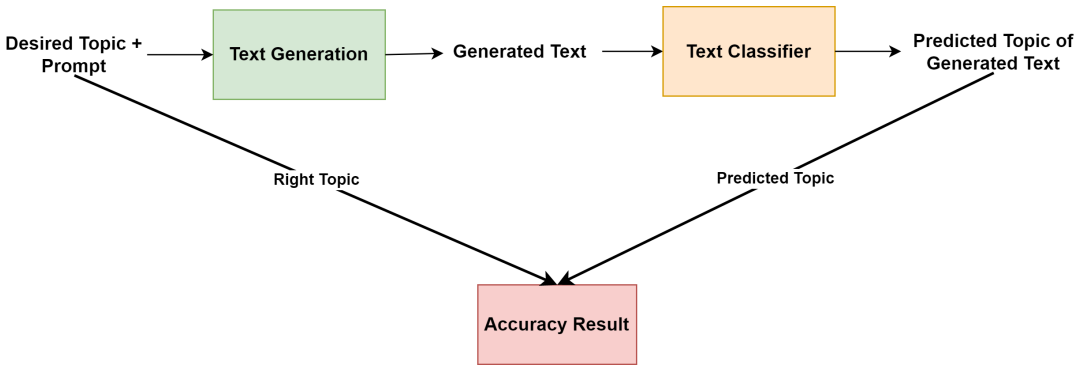

Figure 4.3: Purpose of the text classifier.

topic classifier, we can use the result of the classifier as a quantitative evaluation in addition to the qualitative evaluation made by human. Thus, in this thesis, classification accuracy results are used by comparing the topic that the texts generated on the desired topic should belong to and the topic they actually belong. In order to do this, it is necessary to obtain a reliable multi-topic classifier. As can be seen in Figure 4.3, the topic controlled generated text which is on the desired topic can be evaluated if it is in the desired topic or not with the reliable text classifier. However, this evaluation does not give us information about whether the generated text is understandable, fluent or in accordance with the language rules. It only measures whether it is in the desired topic or not.

To create a reliable multi-topic classifier, three deep learning models are studied by eliminating various models according to the results of some studies in the literature review. As a result of the examination of similar studies in Turkish and English, we make experiments in three models. These are Custom Transformer Encoder, One Dimensional Convolutional Neural Network (CNN), and Fine-Tuned Turkish BERT model (BERTurk).

As the experimental design, TC32 dataset, which is introduced in Chapter 3, is first pre-processed for multi-topic classification models. Then, we explain Transformer Encoder, CNN, Fine-Tuned BERTurk models respectively and the we compare results of experiments .

4.3 Data Pre-processing for Text Classification

Data pre-processing is a very important step for NLP tasks. It transforms text into a more suitable form so that training models can perform better. Therefore, in this section, data preparation processes for text classification models will be explained.

First of all, since the topic names are kept as text, we assign an integer value to each topic. Thanks to this process, we use these integer values, called topic id, as labels during the classification phase. After we apply some data cleaning processes to the reviews. These are:

- Converting all the reviews to lowercase. Thus, Turkish stopwords loaded from the NLTK library [83] could be found in the text.
- Removing Turkish stopwords ('acaba', 'ama', 'aslinda', 'az', 'bazi', 'belki', 'biri', 'birkac', 'birsey', 'biz', 'bu', 'cok', 'cünkü', 'da', 'daha', 'de', 'defa', 'diye', 'eger', 'en', 'gibi', 'hem', 'hep', 'hepsi', 'her', 'hic', 'icin', 'ile', 'ise', 'kez', 'ki', 'kim', 'mı', 'mu', 'mü', 'nasıl', 'ne', 'neden', 'nerde', 'nerede', 'nereye', 'niçin', 'niye', 'o', 'sanki', 'sey', 'siz', 'su', 'tum', 've', 'veya', 'ya', 'yani').
- Removing HTML line-break tags (br /).
- Removing numbers.
- Removing punctuation
- Removing extra spaces between words.
- Converting Turkish characters (ı,ö,ü,ğ,ş) to ı,o,u,g,s for convenience.
- Removing the "devamini oku" (read more) statement at the end of each review mentioned in the Chapter 3.

After the cleaning process there are a total of 427230 reviews and labels in the new dataset, which includes 32 topics. 80% of the whole dataset is divided as all train data and 20% as test data. Then, 10% of all train data is separated as validation data. Sizes of train, validation and test data are :

- Train data size: 307605
- Validation data size: 34179
- Test data size: 85446

The reviews of the train data cannot be given to the model as strings. Therefore, each word in the reviews should be converted to tensors. For this process, first of all, the number of words to be searched should be limited. For this task, we set the vocabulary size as $100k$. In other words, only the first $100k$ words from the current dataset will be considered. In addition, we set the maximum sequence length set as 40. Each sequence will be a maximum of 40 tokens in length and if a review is shorter then padding is applied, how we make this choice is explained in Chapter 3. Using these parameters, we adapt the Text Vectorization Layer according to the reviews of the train data. Text vectorization layer from Keras transforms a batch of strings (one sample = one string) into either a list of token indices (one sample = 1D tensor of integer token indices) or a dense representation (one sample = 1D tensor of float values representing data about the sample's tokens) [85]. In addition, the cleaning operations described above are carried out by updating the *custom standardization* function of the text vectorization layer. Text vectorization layer, that we adapt according to the words of train data, transforms the string inputs passed through it into 40-size tensors of the numbers it is learned. The vocabulary resulting after the adaptation of the text vectorization layer can be summarized for first 10 words as follows:

- 0 represents the word:
- 1 represents the word: [UNK]
- 2 represents the word: bir
- 3 represents the word: gun
- 4 represents the word: aldim
- 5 represents the word: sonra
- 6 represents the word: yok
- 7 represents the word: ragmen

- 8 represents the word: once
- 9 represents the word: tl

We prepare the train, test and validation sets in this way and finalize them by going through shuffle, batch, cache and prefetch processes. Thus, they become usable by classification models. An example input and output is as follows:

- Given raw data:

  başkent doğalgaz kartlı sayacı pili bittiği degiştirdikartlı doğalgaz sayacımın ekran görüntüsü olmadığı başkent doğalgazı arayarak arıza kaydı oluşturdum servis gelir gelmez başkent doğalgaz kağıdını vererek sayaç değişimi yapılacağını 3 gün içerisinde verilen kağıtla başkent doğalgaza gitmem gerektiğini söyledi kağıdı okuduğumda arıza tespitdevamını oku

- Text after tokenized and cleaned:

  baskent dogalgaz kartli sayaci pili bittigi [UNK] dogalgaz sayacimin ekran goruntusu olmadigi baskent dogalgazi arayarak ariza kaydi olusturdum servis gelir gelmez baskent dogalgaz kagidini vererek sayac degisimi yapilacagini gun icerisinde verilen kagitla baskent dogalgaza gitmem gerektigini soyledi kagidi okudugumda ariza

- Tokenized and transformed to a vector of integers: tf.Tensor([[940 343 5246 4465 2957 2593 1 343 9062 244 3688 319 940 3608 951 193 881 1074 40 967 2973 940 343 9747 1469 1349 592 2321 3 185 379 33247 940 23821 4140 522 225 1957 8772 193]], shape=(40,), dtype=int64)

- Output (topic id) : 6

- Output (topic as text): *enerji*

We also prepare the TC5, consisting of the 5 most prominent topics (finans, saglik, turizm, spor, gida) in the same way. Because we will need to observe some problems related to topic sensitivity of the controlled text generation model. In this subset, there are 67432 reviews consisting of 5 different topics. After the classification model is selected according to the experiments conducted on TC32, a classifier is obtained

for TC5 too. The reason for that is, that controlled and uncontrolled text generation experiments will be done using TC5 in the future.

4.4 Multi-Topic Text Classification Models

In this section, experiments on three deep learning based models will be explained in order to create a reliable multi-topic text classifier that will decide whether the generated texts are in the desired topic. Using the prepared dataset, firstly the transformer encoder model, then the CNN model and finally the Fine-tuned Turkish BERT (BERTurk) model will be explained. Although it seems to have given successful results in all three models, the most appropriate model selection will be explained in the results section in terms of the operations performed in this thesis.

4.4.1 Custom Transformer Encoder

Transformer is a deep learning based encoder-decoder structure that has achieved very successful results in the NLP field, announced in 2017 [6]. Transformer, which was first introduced through text translation task, is used effectively in many NLP tasks today, either entirely or only with encoder-decoder blocks. Basically, the biggest change provided by Transformers is that, unlike recurrent neural networks, words can be taken in parallel instead of sequentially, and infer the relationship of each word with each other. For these, position embedding which holds the location information of the tokens and self-attention structures are used in Transformer. Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the sequence [6]. Transformer encoder-decoder structure is shown in Figure 4.4.

The section in the left half of the transformer architecture is the encoder block. The main task of the encoder block is to represent an input sequence in a continuous representation. The decoder can then be fed with this information. The decoder shown on the right is for generating an output sequence. Therefore, as in this thesis, the decoder side is generally used in text generation and will be explained in the text generation section.

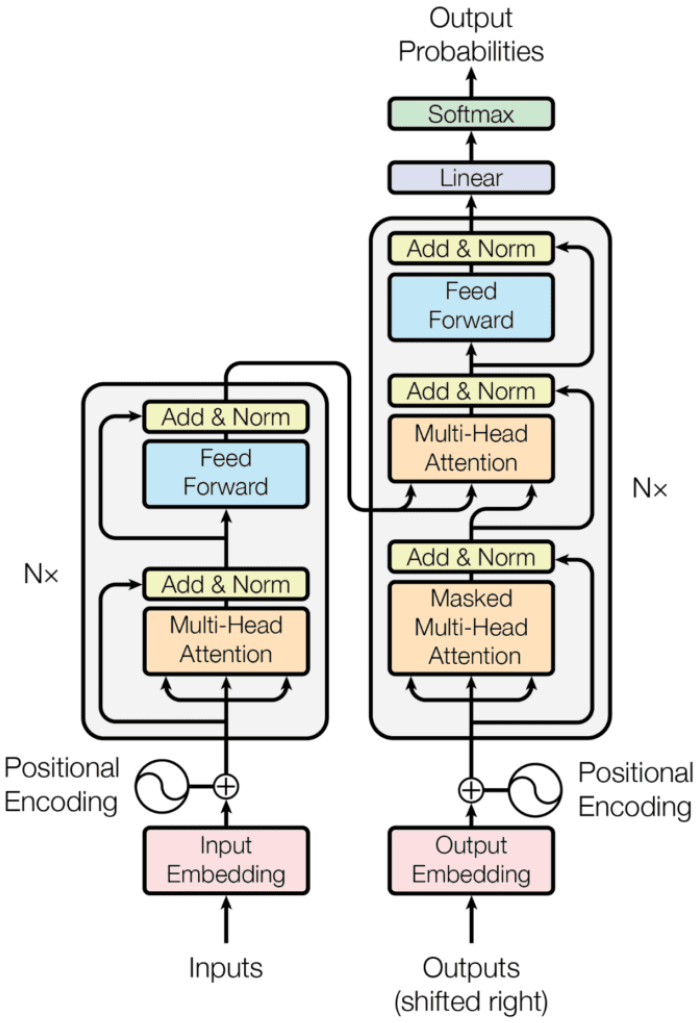

Figure 4.4: The encoder-decoder structure of the transformer architecture taken from [6]

The model we use for text classification task is the Transformer Encoder Classifier. The aim is that the model can learn very well the classes of texts by creating effective representations of texts with known labels. Afterwards, it can make a successful classification in texts it has never seen. First of all, token embeddings and position embeddings of the tokens in the input sequence are prepared. Since the transformer structure provides parallelism by taking the tokens at the same time, it should be fed with the position information of the tokens. In the generic transformer architecture, the encoder consists of a stack of six identical layers. Each layer is composed of two sub-layers, multi-head self-attention mechanism and fully connected feed-forward network. However in this experiment we use single transformer encoder block for

text classification model.

For the model that we prepare in this study, we pass the vectorized input sequence through token and position embeddings and then we give them to a single transformer encoder block. After the transformer encoder block obtains the attention score matrices containing the score of every token with different attention patterns, these inter-layer features are connected into a fully-connected dense layer in order to reach the combination of varying attention scores of the terms [33]. Then, we obtain estimates passed through the pooler, dropout, and the last fully-connected dense layer, and the softmax function based on the number of topics used in the last layer. The architecture of the custom transformer encoder classification model is shown Figure 4.5.

In this model, we set the embedding size for each token as 32, number of attention heads as 2, hidden layer size in feed forward network inside transformer and in the classification layer as 64. Inputs after pre-processing are vectors consisting of the representation of a maximum of 40 words and contain 32 different topics.

As a first step, we obtain the results of *loss* and *sparse categorical accuracy* related to validation by using the train and validation datasets. We also use *EarlyStopping* method to prevent the overfitting problem and to decide on the correct epoch number accordingly. The EarlyStopping method, stops training when a monitored metric has stopped improving [85]. In this study, we select validation loss as the monitored metric and set the patience value to 3. In other words, where there is no improvement in training after 3 epochs, the training is terminated. This model training ends in the 5th epoch and the results are :

- Training time : 3.75 minutes
- Earlystopping epoch (patience 3) : 5
- Test loss : 0.214, Test data Sparse Categorical Accuracy : 0.955

The *loss* and *sparse categorical accuracy* values calculated in each epoch according to the train and validation data are shown in Figure 4.6 and Figure 4.7.

As a second step, we check the model with the entire train dataset (train+validation) using the *K-fold Cross-Validation* method. This method is used to ensure that the

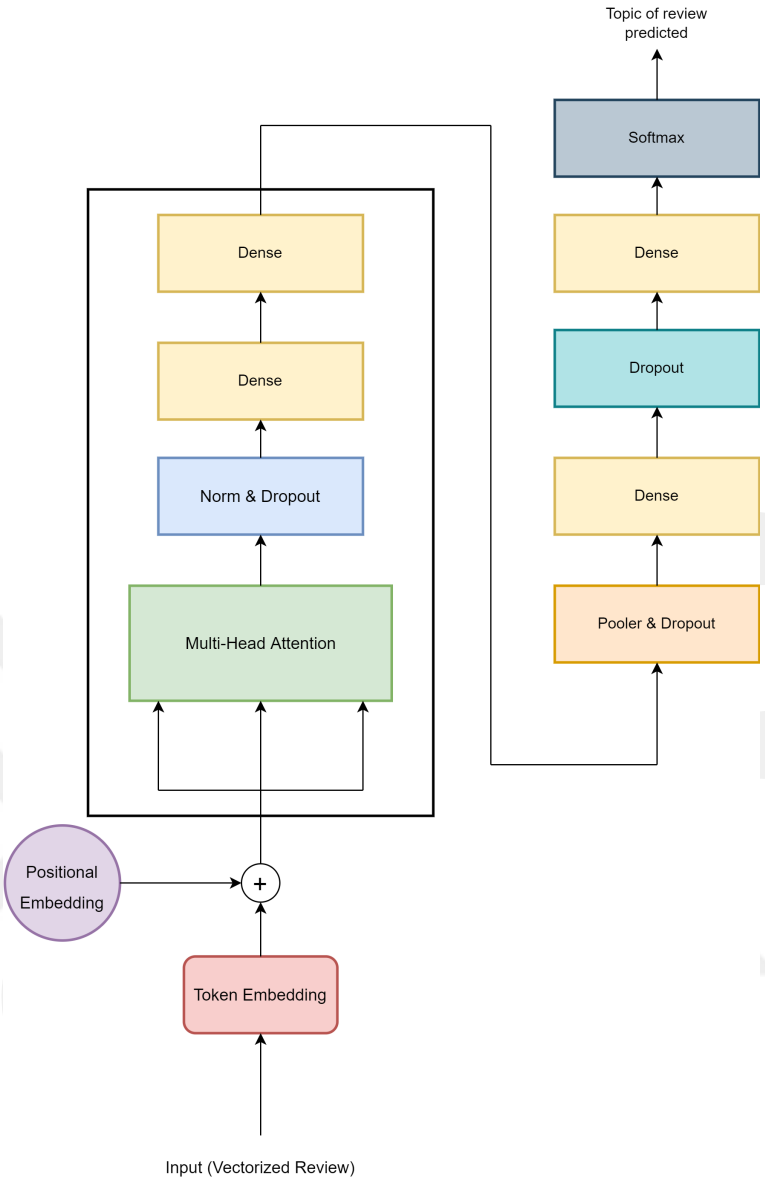

Figure 4.5: The architecture of the transformer encoder classification model.

trained model does not overfit a single train data. A dataset is randomly divided into k disjoint folds that have approximately the same number of instances[86]. In this experiment, we set the k to 5 so that the entire train dataset is split 5 times into different random train and test data and the results of the model are we obtain with 5 different training for 2 epochs. K-fold cross-validation results are shown in the Table 4.1. Accordingly, it is seen that the model achieved the similar evaluation success with 5 randomly selected different datasets.

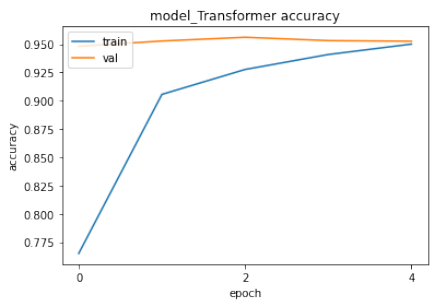

Figure 4.6: Sparse categorical accuracy of the transformer encoder classification model.

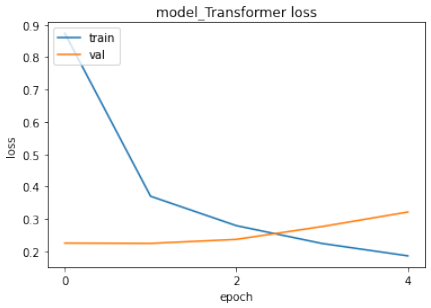

Figure 4.7: Loss of the transformer encoder classification model.

Table 4.1: K-fold cross-validation results for transformer encoder classifier.

k	Test Data Loss	Test Data Sparse Categorical Accuracy
1	0.215	0.955
2	0.210	0.956
3	0.207	0.957
4	0.207	0.954
5	0.221	0.953

Finally, we evaluate the model according to traditional classification metrics. For this we train the model with all train data and obtain metrics according to the test data. These classification metrics are *precision*, *recall* and *f1-score*. In the classification problems, predictions are labeled in four ways to look at evaluation metrics. These are True Positive (TP), False Positive (FP), True Negative (TN), and False Negative (FN). The *precision* is the fraction of true positive elements divided by the total number of positively predicted units [87].

$$Precision = \frac{TP}{TP + FP} \tag{4.1}$$

Recall measures the model's predictive accuracy for the positive class: intuitively, it measures the ability of the model to find all the Positive units in the dataset [87].

$$Recall = \frac{TP}{TP + FN} \tag{4.2}$$

F1 score aggregates Precision and Recall measures under the concept of harmonic mean [87].

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{4.3}$$

For multi-class classification tasks, these metrics should involve all the classes from the dataset. Therefore the weighted average of *precision*, *recall* and *f1-score* metrics for each of the 32 topics is calculated. Weighted average is found by weighting the mean of precision, recall, f1 values calculated for each class against the support, which is the actual number of samples for each class (85440 for us).

- Weighted Average Precision = 0.958
- Weighted Average Recall = 0.958
- Weighted Average F1 = 0.958

As a result of the latest model training, we obtain values below according to the test data:

- *loss* = 0.193
- *sparse categorical accuracy* = 0.958

According to these results, we can say that a single block custom transformer encoder classification model is quite successful in classifying the consisting 32 topics reviews.

4.4.2 Convolutional Neural Network (CNN)

Although Convolutional Neural Network (CNN) is generally used in image processing in research, it is also frequently used in text classification problems. CNNs classify text by working through the following steps [88] :

1. 1-dimensional convolving filters are used as ngram detectors, each filter specializing in a closely-related family of ngrams [89].

2. Max-pooling over time extracts the relevant ngrams for making a decision [89].

3. The rest of the network classifies the text based on this information [89].

In this study, vectorized reviews of the train data for the CNN model are layered through embedding. We set the embedding size to 32, hidden layer size for feed forward network to 64. The model is prepared with 2, 1D Convolutional layers, 1D MaxPooling, 1 Flatten and 2 fully-connected dense layers. MaxPooling1D layer downsamples the input by taking the maximum value and flatten turns the pooled feature map into a single column that will be passed to the fully connected layer [90]. The summary of the CNN model is shown in Figure 4.8.

The same experiments that are done for the transformer encoder classification model are also performed for the CNN classification model. Accordingly, first, we train the model using train and validation data to determine and control parameters. In this training, we use EarlyStopping method by choosing patience values is 3. Considering the validation loss according to this method, there is no further improvement, so the training ends at the 12th epoch. It is seen that this model needs more epoch for training than transformer encoder. According to the test data of this training, the results are as follows:

- Training time : 4.27 minutes
- Early stopping epoch (patience 3) : 9
- Test data loss : 0.296
- Test data Sparse Categorical Accuracy : 0.936

The *loss* and *sparse categorical accuracy* values calculated in each epoch according to the train and validation data are shown in Figure 4.9.

Then, we apply the K-fold cross-validation method for this model. Thus, we obtain the results of 9 epochs of training over 5 different randomly divided trains and test data of the model. In this way, we ensure that the model does not overfit a single dataset. K-fold cross-validation results are shown in Table 4.2. Accordingly, it is seen that

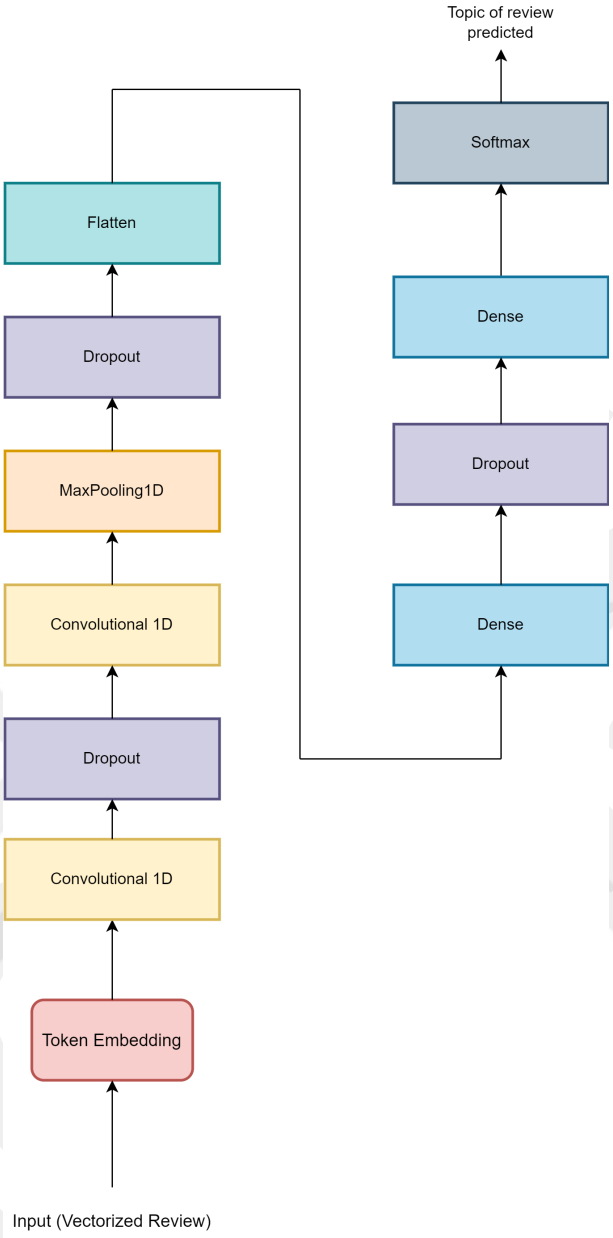

Figure 4.8: The summary of the CNN classification model.

the model achieved a similar evaluation success with 5 randomly selected different datasets. We can say that the CNN model is close to the transformer encoder model but slightly less successful in the multi-topic text classification task for the TC32 dataset. However, it can be said that both models are quite successful in correctly classifying topics of the reviews.

Weighted average values of classification metrics for CNN model using each classe :

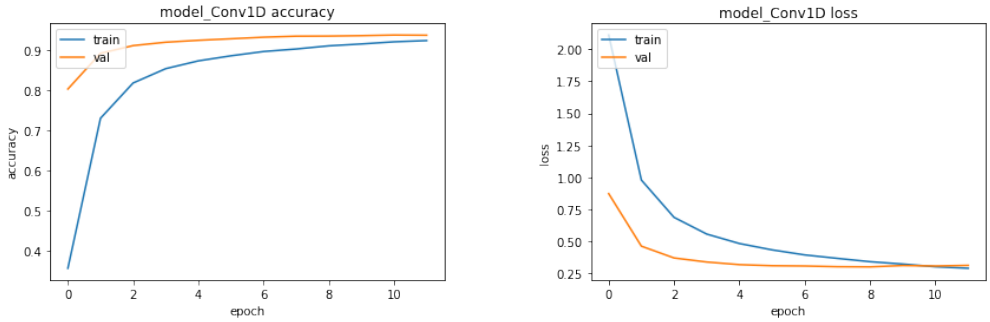

Figure 4.9: Sparse categorical accuracy and loss of the CNN classification model.

Table 4.2: K-fold cross-validation results for CNN classifier.

k	Test Data Loss	Test Data Sparse Categorical Accuracy
1	0.302	0.936
2	0.253	0.946
3	0.243	0.948
4	0.245	0.948
5	0.244	0.946

- Weighted Average Precision = 0.958
- Weighted Average Recall = 0.958
- Weighted Average F1 = 0.958

As a result of the latest model training, we measure according to the test data :

- *loss* = 0.272
- *sparse categorical accuracy* = 0.941

4.4.3 Turkish BERT Model (BERTurk)

BERT, which stands for Bidirectional Encoder Representations from Transformers is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers and it uses masked language models to get pre-trained deep bidirectional representations of texts [8]. BERT model

contains 12 layers of 768 hidden size and 12 self-attention heads [91]. It basically includes multi-layer bidirectional transformer encoders. Pre-trained BERT model consists of a trained set of bidirectional transformer encoders that are frequently used in text classification problems. However, since the model itself is produced for the English language, it must be fine-tuned to be used in other languages. For this reason, we use the BERTurk model, which is pre-trained in Turkish for this study.

BERTurk is a BERT model for Turkish language. The model is trained using many Turkish datasets (Turkish OSCAR corpus, Wikipedia dump, various OPUS corpora etc.) and the final training corpus has a size of 35GB [27] [28].

Uncased BERTurk model used in this thesis is taken from HugginFace [27] and fine-tuned with TC32 dataset. Uncased means that the data used in training this classification model has been trained by converting it to lowercase. Since we prepared our data in that way, it would be correct to use this pre-trained model. Also, we get help from the SimpleTransformers [92] library that allows us to easily use the pre-trained models from HugginFace platform [93].

BERTurk includes its own trained tokenizer. Therefore, the input must be in the string data structure. In other words, only the texts are cleaned without the tokenization and vectorization sections described in the data pre-proccesing section. Training and saving of the BERTurk model with the entire TC32 dataset cause a memory shortage on Google Colab, so it can be saved with $200k$ sample data, to includes reviews from every 32 topics in a balanced way. Thus, the cleaned and lowercase reviews from this sample train data are given to the BERTurk uncased model as a string, and the model is trained with 1 epoch. The reason for choosing one epoch here is that epochs take too long. In addition, although one epoch is trained in the experiments, it is seen that the success of the model is very high. Thus, the pre-trained BERturk model is fine-tuned with sample of TC32. According to the test data of this training, the results are as follows:

- Training time : 82 minutes
- Test data loss : 0.127
- Test data Sparse Categorical Accuracy : 0.967

Weighted precision, recall, f1 metrics we obtain by the Fine-tuned BERTurk classification model using each class :

- Weighted Average Precision = 0.968
- Weighted Average Recall = 0.968
- Weighted Average F1 = 0.968

According to the classification report of this model, more than 90 percent of success is achieved in all 32 classes. The most important reason for this is that the BERTurk model is trained using big Turkish corpus. The model learned very well about the representations of tokens during this training. It is also an advanced model when it uses a multi-layer mask transformer encoder structure. In this way, it is very successful in understanding and distinguishing Turkish texts. Although we have given much less training data than other models, it successfully classifies the fixed sized test data we use.

Due to all these experiments, the fine-tuning of the BERTurk model takes much longer than other models and it requires extra hardware resources to save the model using all of the data. Although fine-tuning the BERTurk model gives the highest results in terms of classification metrics, it takes much longer to train compared to the other two models and a lot of memory to upload the pre-trained model weights, fine-tune and save them for our experiments. For all these reasons we cannot use all TC32 reviews for this model.

However, our lack of resources does not affect the success of this model. Fine-tuning pre-trained models or training from scratch using the same advanced architecture with a different dataset when the necessary conditions are met is very useful for many NLP tasks.

4.4.4 Results

The results of the experiments, in which the three text classification models are compared, are shown in Table 4.3. Accordingly, all three classification models gave higher

Table 4.3: Last results for three classification models.

Classification Model	Sparse Categorical Accuracy	Precision	Recall	F1-score
Transformer Encoder	0.958	0.958	0.958	0.958
CNN	0.941	0.943	0.942	0.942
Fine-tuned BERTurk	0.967	0.968	0.968	0.968

results on test data than traditional classification metrics. This indicates that eventually all three models can be used in a topic classification task. Accordingly, all three classification models yield higher results on test data according to traditional classification metrics. This shows at the first observation that all three models can be used in the topic classification task discussed in this thesis, but we actually have some implications.

Although all three models give close and successful results at the end of the experiments, some issues has arisen that needs to be examined here. The first thing that draws attention here is that fine tuning the BERTurk model requires much more hardware resources to do it with all the data. According to this situation, there is also the possibility of creating a great need for time. Despite the use of sample data, the train process took much longer than the other two models. This can cause problems in training and saving the model.

As a second observation, there is a possibility that the success of these models, which give similar results, will only occur when sufficient data is available for the model. For this reason, it is desired to observe how the success of the models would be when there are much fewer data. Accordingly, while the test dataset is fixed (85447 reviews), 1%, 3%, 5%, 10%, and 20% of train data from the whole dataset are split and the results are compared by training all three classification models with these train sets. The F1 score results obtained are shown in Table 4.4. According to this experiment, it has been seen that the CNN classification model actually needs much more data than other models in order to achieve a certain level of topic classification success. The fine-tuned BERTurk model shows successful results even with very little data, but as the data increases, the required time and hardware power also increase. Moreover, even with very little data, the process of loading weights takes longer than other models.

Table 4.4: F1 score of three classification models with different amounts of train data.

	CNN	Transformer Encoder	BERTurk
%1 of train data (3417)	0.003	0.330	0.665
%3 of train data (10253)	0.129	0.854	0.903
%5 of train data (17089)	0.428	0.885	0.929
%10 of train data (34178)	0.698	0.910	0.946
%20 of train data (68356)	0.862	0.931	0.958

Considering all these observations, we choose the Transformer Encoder classification model to be used in the task of determining the topic of the generated reviews. For this reason, we train this model using the entire TC5 dataset and store for later use. Thus, when we generate controlled multi-topic reviews, we can measure whether the reviews generated are on the desired topic by using this classification model that we train and save.

CHAPTER 5

TEXT GENERATION WITH A SINGLE GPT BLOCK

In this section, firstly, the text generation task in general and GPT architecture for text classification task will be explained. Then, the data pre-processing, training and generating text stages for the model we use in our experiments and the evaluation metrics we use in the evaluation of the generated texts will be explained.

5.1 Text Generation Task

As mentioned in the Chapter 1, text generation is an NLP task that automatically enables computers to generate text in human language. The models used in text generation are called language models (LM). The language model we use in this thesis is known as the Causal Language Model. Accordingly, the model tries to predict the next token based on the given input. The language model mentioned in this thesis should always be considered as a causal language model.

The purpose of a causal language model is to predict the next token based on a given input. Language models that uses transformer decoder blocks like GPT, outputs one token at a time. After each token is generated, that token is added to the sequence of inputs and the new sequence becomes the input to the model in its next step [94]. Therefore, these models called auto-regressive models. In this way, the model can learn to generate accurate texts each time using the information it learns to predict the next token. If pre-trained models are not used, this information is taken into account when preparing the data to be given to the language model manually. Basically the example input and output with using TC32 for the LM are as follows:

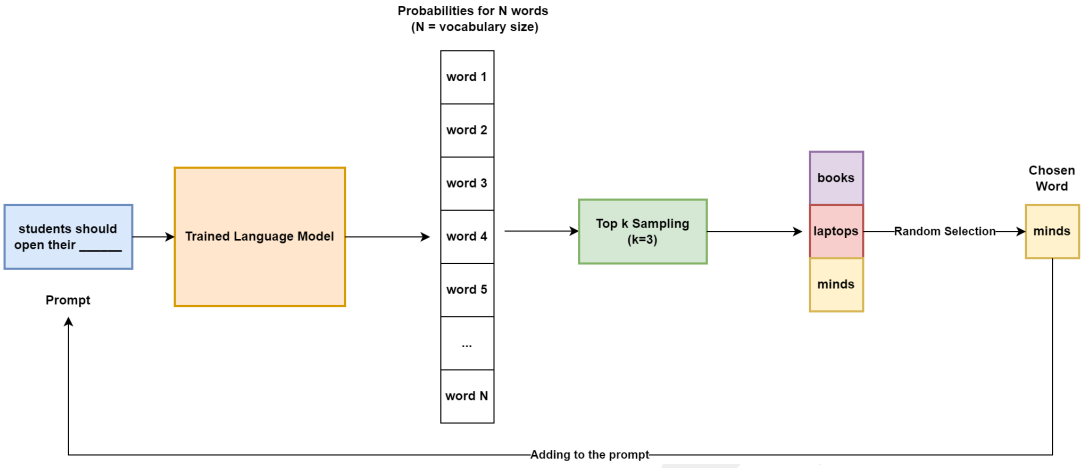

Figure 5.1: Illustration of text generation task

- input (in text): ozel genesis hospital randevu sorunu diyarbakir ozel genesis hastanesi randevu sorunu yasatiyor parayla dustugum duruma
- output (in text): genesis hospital randevu sorunu diyarbakir ozel genesis hastanesi randevu sorunu yasatiyor parayla dustugum duruma bak

In text generation task, a prompt, sometimes can be just a ”start” command or a specific token, is given to the trained language model. As the language model learns to find the probabilities of the next token as it is trained, it calculates the probability for each token in the vocabulary. Then a sampling method like *Top-k* sampling that also we use, takes the highest k of these probabilities and chooses one of them randomly. Random selection could be beneficial in increasing diversity in the final stage. Thus, the selected token is added to the prompt and text generation continues in this loop structure. This process is illustrated in Figure 5.1.

In this section, the first language model we used to construct the uncontrolled text generation task, a single layer causal GPT transformer language model and experiments will be explained. Since the purpose of the studies described in this section is only to understand the task of generating text, the texts created are random, not on the determined topics. Understanding the issue of text generation plays a very important role in the creation and experimentation of changes that can be made in the language model, in other words, controlling techniques that can be applied.

5.2 GPT Architecture for Text Generation

In this thesis, we use the GPT (Generative Pre-trained Transformer) [49] causal language model structure, which was first released by OpenAI in 2018. GPT models are basically transformer based models that pre-trained using huge text data to generate any type of text like human text, or programming code.

It has been gradually developed over time and presented as more than one model as GPT, GPT2, GPT3. These pre-trained models have been quite successful in generating texts in the English. Of course, very large hardware resources have been used for the training of these multi-layered and have so many trainable parameters models. Since most researchers do not have such an opportunity, they can usually obtain their own language models by fine-tuning these models with different data or by training the same architecture with much less data than the original. The HuggingFace platform [93] is very useful for these mentioned methods. On this platform, pre-trained language models can be fine-tuned with different datasets or trained from scratch. In addition, newly acquired models can be stored on the platform, shared and used for various NLP tasks such as text generation. We will use the methods mentioned here in future experiments, needing a more advanced language model. However, firstly, we use the Miniature GPT [15] model understand the structure of the GPT and text generation task. Miniature GPT includes a custom GPT transformer decoder block. It is simplest GPT model but while creating this model, every step of the transformer decoder structure is created in a clear and accessible way, so it is useful for controlling experiments. The actual GPT (Generative Pre-Training) models are very advanced and trained with very large datasets. But it's basically based on the same transformer decoder only structure. The reason why the experiments are carried out with Miniature GPT model at the beginning is that every point of the architecture is open to access and it is simple compared to the pre-trained language models. Below parameter and transformer decoder block numbers of the original GPT models and the Miniature GPT model are listed:

- GPT : 12 blocks transformer decoder, 117 million parameters, with 7k unique unpublished books

- GPT-2 : 36 blocks transformer decoder, 1.5 billion parameters, with 40GB text data
- GPT-3 : 96 blocks transformer decoder, 175 billion parameters, with 45TB text data
- Miniature GPT : 1 block transformer decoder, 40 million parameters, 67k reviews (TC5).

Accordingly, Miniature GPT has fewer parameters, fewer data, and fewer decoder blocks, the Turkish texts generated with this model can be weak in terms of semantics and grammar compared to the pre-trained GPT models. It is therefore probably more beneficial to use multi-layer models. However, it would be difficult to build so many decoder structures by hand, instead, it is easier to train pre-trained models from scratch. But in the first stage, we chose to create a single-layer GPT block to explore the places where modifications can be made in the generation of controlled text and to access every module of the decoder structure. This model is quite convenient in terms of understanding the structure of text generation and facilitating the application of controlling techniques to be applied to a language model.

The structure of the causal language models is based on the transformer decoder. If we remember a generic transformer, the part on the right is the decoder section and is used to generate tokens. Generic transformer structure is shown in Figure 5.2.

Transformer decoder part is similar to the encoder part. The input text is first pass through the Token and Position Embedding Layer. Token embedding is a vector using floating point values which are trainable parameters to represent each token. This learned representation of tokens based on their usage allows tokens with a similar meaning to have a similar representation. Token embedding is used to learn the best representation of tokens based on their context during the training. Also, in the transformer, positional embedding is injected into each token embedding so that the model can know token positions without recurrence. Briefly, embedding layer try to find best representation of tokens with their positions during the training. The most important difference of the decoder structure from the encoder is the masked multi head-attention section which is stacked masked-self attentions. Normally, self-attention

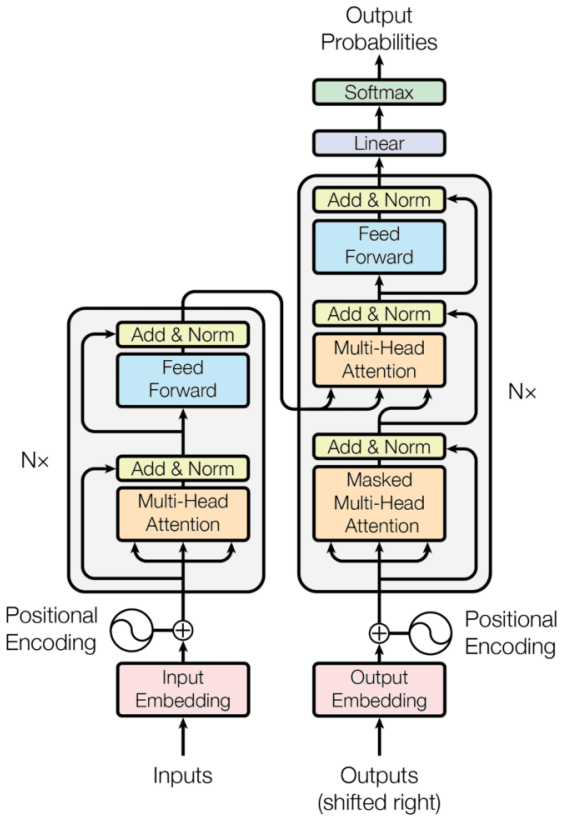

Figure 5.2: The encoder-decoder structure of the transformer architecture taken from [6].

can look at all the tokens that come after the current token. However, masked self-attention does not allow it. In other words, attention cannot see the tokens that come after the current token for that moment. In an auto-regressive model, the tokens will be generated one by one means the token that comes after a current token will be generated later. Therefore, the token at that moment has access only to itself and to tokens generated before it. So that the next tokens are masked. This prevents flow of information from future tokens to current token. The difference between the self-attention and masked self-attention is shown in Figure 5.3. The decoder part passes the output of token and position embeddings through the masked multi-head attention, and then passes it through multi-head attention and feed forward neural network. Then after a linear layer and *Softmax* function it get the final probabilities of the tokens.

A slightly modified version of the transformer decoder structure is used in GPT models. That's why it is referred to as the GPT kind transformer structure. GPT kind

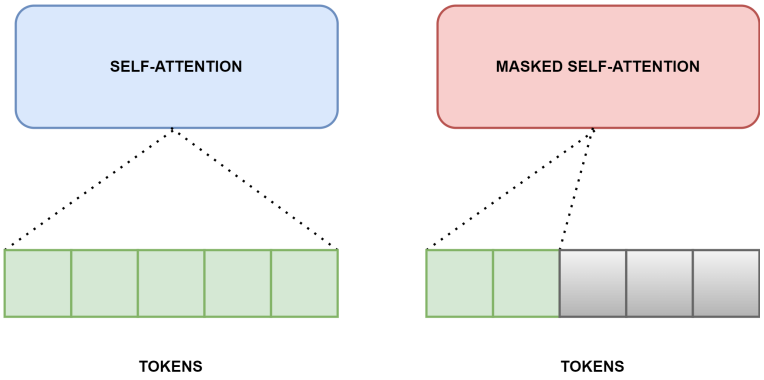

Figure 5.3: Difference between self-attention and masked self-attention

transformer block [95] have used the transformer decoder structure as a language model, but unlike the original one, the second multi-head self attention layer has been removed. In this structure, after token and position embedding, masked multi-head attention is applied to the inputs. After the normalization and feed forward layers, the *Softmax* activation which is a function that converts all the input token probabilities from (-∞,∞) to (0,1) has applied. This layer is necessary to convert the output of the above layers into the actual token probabilities across the entire vocabulary. Thus, the model learns to predict the probability distribution for the next token. This structure is shown in Figure 5.4.

Once the causal GPT language model has been trained, it can be used to generate new texts. For this, the model is fed with a starting token (prompt) and finds the possibilities for the next token. However, at this point, it is necessary to choose next token among the possibilities. This is called sampling. Sampling means choosing the next token based on a conditional probability distribution. After constructing a probability distribution over the vocabulary for the given input sequence, how to choose the next token from this distribution is very important for the text generation task. There are several methods for sampling in text generation, such as: greedy sampling (maximization), temperature sampling, top-k sampling, top-p sampling, beam search etc. The *Top-k* sampling method is used as in the GPT models. *Top-k* sampling takes a random sample from the k most likely candidates from the distribution. *Top-k* sampling ensures that less likely tokens do not have a chance, while also giving other high-scoring tokens a chance to be selected (as opposed to greedy sampling). It has been seen to

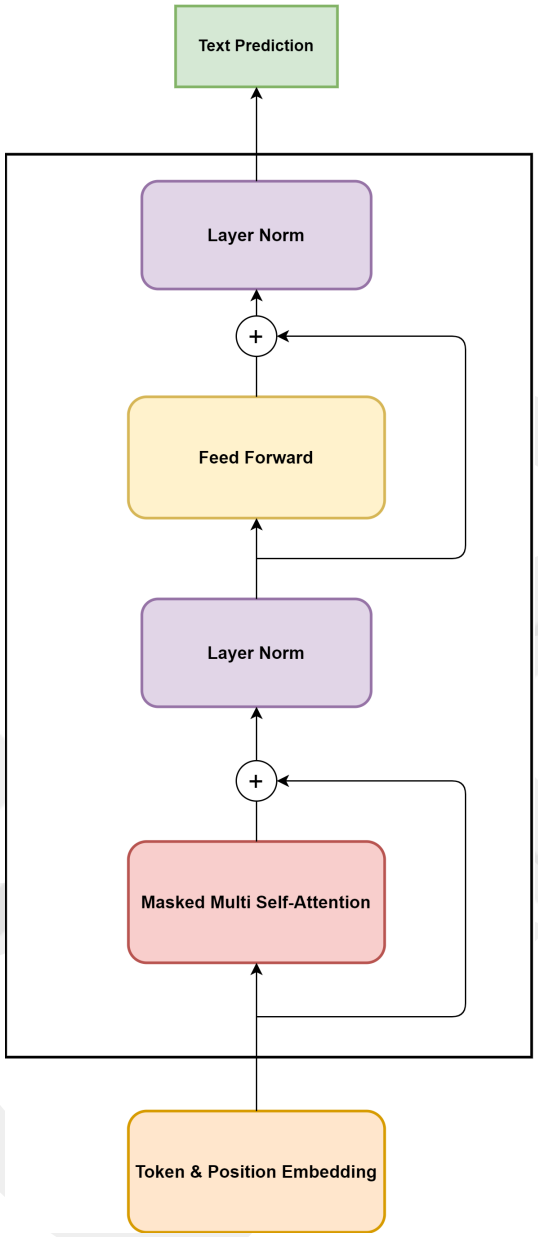

Figure 5.4: The GPT structure taken from [7].

increase generation quality in many scenarios.

5.3 Data Pre-processing for Text Generation

For the uncontrolled text generation experiment with Miniature GPT model, we use TC5 dataset which has 5 topics (finance, health, tourism, sport, food) with 67k Turkish reviews. Each topic contains approximately 13k reviews. This dataset is split from

the TC32 dataset which is discussed in Chapter 3.

In this section, it will be explained how the text inputs to be given to the causal GPT language model to be trained are prepared. Although we make some changes in the input preparation in the controlled text generation part, which will be explained later, the steps for preparing the text input described in here are generally the same. While preparing the dataset for text generation, all reviews are used for training, as there is no need for split such as train, validation and testing. All reviews go through the same cleaning processes mentioned in text classification task, but this time we do not remove Turkish stopwords because they are necessary for the semantic integrity of the generated text. So, vocabulary size is chosen as $100k$ and maximum sequence length is 40. Texts are vectorized using the *Text Vectorization Layer*. A sample input review processed is as follows:

- Given raw data:

  Kanguru Matematik Online Deneme Sınavı!,Kanguru Matematik online sınav için başvuru yapamıyoruz verilen WhatsApp hattına da dönen olmadı. Okuldan kaydımızı yapmıştık ancak ne yazık ki sınav giriş sayfasından kaydımızı tamamlayamıyoruz. Problemi nasıl çözeceğimize dair bilgi de alamıyoruz çünkü tüm giriş siteleri 404 hata kodu veriyor.Devamını oku

- Tokenized and Transformed to a vector of integers:

  5240 3333 253 1626 6331 1 3333 253 2455 5 639 2592 456 886 4239 7 1878 186 11901 14914 10529 56 10 405 61 2455 386 2751 14914 68108 395 69 1 734 117 6 1711 449 188 386 6187

- Text after Tokenized and Transformed:

  kanguru matematik online deneme sinavi [UNK] matematik online sinav icin basvuru yapamiyoruz verilen whatsapp hattina da donen olmadi okuldan kaydimizi yapmistik ancak ne yazik ki sinav giris sayfasindan kaydimizi tamamlayamiyoruz problemi nasil [UNK] dair bilgi de alamiyoruz cunku tum giris siteleri

The language model learns to guess the $(i+1)$th word by looking at all the words up to the ith word. Therefore, the output should be prepared as a one position word-shifted

version of the input. So the Model can use all the words up to the *i*th word to predict the next word [15] . As a result, the prepared inputs and outputs are as follows:

- input (text):

  kanguru matematik online deneme sinavi [UNK] matematik online sinav icin basvuru yapamiyoruz verilen whatsapp hattina da donen olmadi okuldan kaydimizi yapmistik ancak ne yazik ki sinav giris sayfasindan kaydimizi tamamlayamiyoruz problemi nasil [UNK] dair bilgi de alamiyoruz cunku tum giris

- input (vectorized):

  5240 3333 253 1626 6331 1 3333 253 2455 5 639 2592 456 886 4239 7 1878 186 11901 14914 10529 56 10 405 61 2455 386 2751 14914 68108 395 69 1 734 117 6 1711 449 188 386

- output (as text):

  matematik online deneme sinavi [UNK] matematik online sinav icin basvuru yapamiyoruz verilen whatsapp hattina da donen olmadi okuldan kaydimizi yapmistik ancak ne yazik ki sinav giris sayfasindan kaydimizi tamamlayamiyoruz problemi nasil [UNK] dair bilgi de alamiyoruz cunku tum giris siteleri

- output (as vectorized):

  3333 253 1626 6331 1 3333 253 2455 5 639 2592 456 886 4239 7 1878 186 11901 14914 10529 56 10 405 61 2455 386 2751 14914 68108 395 69 1 734 117 6 1711 449 188 386 6187

The train data, which consists of the final inputs and outputs forms of reviews, is ready to be given to the language model after shuffle, batch, cache and prefetch operations.

5.4 Training Language Model and Generating Uncontrolled Text

In this section, the training and results of the language model we use for uncontrolled text generation, which contains single GPT block are explained. This model called Mİniature GPT as we mentioned before. The model is train by combining the token and position embeddings of the prepared data, passing it through the GPT block and

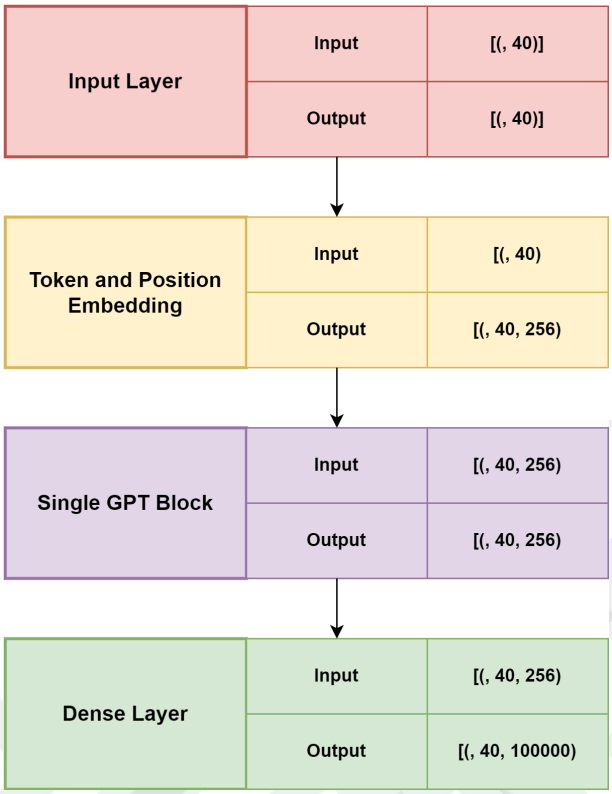

Figure 5.5: Language model summary for uncontrolled text generation.

then through the dense layer. Maximum length for sequences is 40, vocabulary size is 100k, embedding size for each token is 256 and hidden layer size in feed forward network inside GPT block is 256. This model is trained for 5 epochs with 2 attention heads. The summary of the language model that we use for uncontrolled text generation task is shown in Figure 5.5.

During the text generation which is called inference stage, the next token is determined by using top-k sampling. Since we set k as 10, the 10 tokens with the highest probability are selected and then one of them is selected randomly as the next token. Thus, the randomness increases to avoid the repetition of the same tokens. Some of the texts generated uncontrolled using the Miniature GPT and sampling with the top-k sampling are shown in the Table 5.1.

See Appendix A for more uncontrolled generated reviews.

Table 5.1: Samples of uncontrolled generated reviews with the Miniature GPT.

Generated Review
pastacilik yapiyorum ve etkilenip ay oldu ay oldu hala ses seda yok musteri hizmetlerini ariyorum surekli mesgul caliyor bu nasil bir calisma sistemidir anlamadim bir daha da almayacagim lutfen yardimci olun
butik otel de netlesemeyen rezervasyon islemi internet arayisim sonucu internet aradim oranin cagri merkezindeki m s hanim kendisinin oranin telefonu acti gayet icten ve samimi bir anlatim sonucu tam bir anlatim sonucu benimle hemen odeyecek gayet kibar
eczanesi ankara maske sorunu ankara esentepe eczanesi ne gelen eldiven var maskem olmadigi icin onlardan maske istedim ama maske yoktu alamadim sistemde ama ilac almak istedim eczane maskeyi vermediler telefonla bilgi ve maske istedim maskem alamadim eczane bulamadim
bugdayin tam kalmasi levent ten eti burcak aldigimda bugdayin biskuvinin tam ortasinda kalmasi hic yakismadi sana eti gerekli aciklama bekliyorum eti gerekli aciklamayi bekliyorum muhakkak

5.5 Evaluation Metrics and Results

Some evaluation metrics are proposed to be used at some text generation tasks such as text translation and caption generation. At these tasks, there can be a known correct can be compared with the generated text. For example, since a text, that is automatically translated from English to French is also known in French the quality of the generated translation can be found numerically. Also in the caption generation task, the right caption can be determined by a human. However, measuring the quality of the text generated in the field of creative writing and presenting them with reliable metrics is a difficult issue. This problem is a research topic that needs to be developed. In this case the most reliable evaluation in creative text generation is human judgment. To understand whether the generated text is in accordance with the characteristics and rules of that language, in the correct grammatical rules or is comprehensible is for people to read the generated text and make a judgment.

However, some evaluation metrics are still reported in creative text generation studies in addition to human judgment. The most well-known of these is BLEU [12]. In order to calculate this metric, the generated text and the referenced text are needed. BLEU is a precision focused metric that counts n-gram overlap of the reference and generated texts. Basically, it checks whether the generated n-grams occur in the reference text,

so it ignores the structure and semantic information. If two sentences or documents match perfectly, the value of BLEU is 1.0, if there is no overlap between them at all, the BLEU value is 0.0. While calculating the BLEU, first the n-gram matches are computed sentence by sentence, next, the clipped n-gram counts are added for all the candidate sentences and divided by the number of candidate n-grams in the reference corpus to compute a modified precision score, for the entire reference corpus [12]. In this thesis, we examine BLEU for the uncontrolled generated reviews using the Miniature GPT [15] language model. We measured the BLEU score as 0.1802 for 31000 uncontrolled generated reviews, using TC5 as a reference corpus.

Since the BLEU score does not take into account the meanings and relationships of words, another evaluation metric we use to evaluate the generated texts is BERTScore [13]. Although the purpose of this metric is to be used in tasks such as translation and image captioning, just like BLEU, it can give a more reliable result in generated reviews, since it also considers semantics. To retrieve the true semantics of a sentence, BERTScore leverages Transformer based model BERT embeddings. Word embeddings are learned dense token representations [13]. These token representations are stored in a pre-trained model after the training. Since this study is conducted in Turkish, token embeddings of the BERTurk model are used to calculate BERTScore. BERTScore computes the similarity of two sentences or documents as a sum of cosine similarities between their tokens' embeddings. The cosine similarity between two vectors (representations two sentences on the vector space) is a measure that calculates the cosine of the angle between them. In contrast to string matching (e.g., in BLEU) it computes similarity using contextualized token embeddings of trained models. The outputs of the score function are tensors of precision, recall, and F1. Complete score matches each token in x to a token in $\hat{x}$ to compute recall, and each token in $\hat{x}$ to a token in x to compute precision [13]. Precision, recall, and F1 values can be taken as an evaluation, but according to the BERTScore article, it is recommended to use the F1 score. We measure F1 score for 31000 uncontrolled generated reviews with Miniature GPT language model as 0.42.

CHAPTER 6

CONTROLLED MULTI-TOPIC TEXT GENERATION

Controllable Text Generation (CTG) is the task of generating text according to the given control element [9]. This control element can be many things like politeness, sentiment, formality, topics, keywords etc. In this study, we aim to create topic control while generating texts. Accordingly, while a language model (LM) generating the meaningful text, the generated text should be on the desired topic.

Controllable text generation task can be formalized mathematically in a unified form, given a vocabulary V, the goal of CTG is to generate a target text $Y = (y_1, y_2, \ldots, y_n)$, where $y_n \in V$ with respect to a control element denoted as C which is the topic in this task [96]. CTG can be formally described as:

$$P(Y|C) = p(y_1, y_2, \ldots, y_n|C) \tag{6.1}$$

As for the sentence Y generated by the model, it is also expected to satisfy the constraint conditions while conforming to the general natural language characteristics such as fluency, rationality, and readability, to the greatest extent [96].

Depending on the structure and features of the LM used in the controlled text generation, we apply some changes to the LM at different stages. Basically, these changes can be made during the training, during the text generation after training which is called inference, or at both stages.

Firstly, we can modify some part of the LM during the training as follows. The control element can be added to the decoder's sequential input or it can be added to the

encoder's output which is called external input. During the training of the language model changes can be made in the model structure, for example additional gates can be added to the RNN model. Also, a control element can be added by modulating the sequential output at each time step, before projecting it to the vocabulary space, like adding the attention mechanism. Finally, the loss function used can be modified to compare the difference between the text generated and the desired control during training [9].

Secondly, control can be achieved at inference time by implementing some token selection sampling techniques. The modified sampling technique helps choosing the appropriate token for the desired control. Controlled text generation can be achieved by using one or more controlling techniques together.

This chapter will first describe controlled multi-topic text generation experiments using the single-layer GPT LM called Miniature GPT described in Chapter 5. The reason why we work on this model, in the beginning, is that it is a simple and fast language model that can be accessed to every module of the GPT decoder structure. In this way, we are able to easily observe the modules where the control element can be added, in other words, the modifications can be made, and we able to do our experiments accordingly. We study three different controlling techniques on this model. In general, these techniques can be summarized as follows. The first two are modifications make to the sequential input at each time step during the training of the LM. The third one is the modification process at the inference phase of the trained LM by using a text classification model as a sampling strategy. This technique that we propose, is called *Sampling with a Topic Selection Classifier*.

After three controlling techniques introduced using the Miniature GPT LM, experiments using a more advanced language model will be described. Since the controlled multi-topic generated texts obtained from the Miniature GPT LM, they are poor in terms of grammar and meaning, we conduct text generation experiments with a more advanced language model. For this, we train a multi-layered Causal GPT LM architecture from scratch with TC32 called Multi-layer GPT. Compared to the first one, we observe that the generated texts are grammatically smoother and more meaningful with this model because of the learning capacity. Since the success of this model in

generating meaningful text shows dependence on the prompt given at the beginning, we create a prompt generator model. Thanks to this model, we have the text generated by accepting the first prompt created on the desired topic as a start. This provides the first control over this model. Then we apply the *Sampling with a Topic Selection Classifier*, which we propose, and our main technique for controlling. This section summarizes the experiments in detail.

6.1 Miniature GPT

In this section, we apply the controlling techniques to the Miniature GPT LM. The first experiments with this simple LM are very important in terms of understanding the model's structure and controlling techniques. For this task, we use the TC5 dataset, which consists of 5 topics, as in uncontrolled text generation in Chapter 5.

We add each of the controlling techniques applied, including training and post-training of the model, on top of the previous one. These techniques are:

- Modifying Sequential Input with Topic
- Modifying Sequential Input with Keywords
- Sampling with Topic Selection Classifier

The first two of these techniques are the techniques created by adding the controlling element to the sequential input at each time step during the training of the model. The last technique that we propose, however, is about modifying the token selection part, it is a new strategy of sampling from probability distributions.

The most important evaluation that we use to measure the quality, grammar, and meaningfulness of the generated texts is human judgment. In addition, we investigate by using the BLEU and BERTScore metrics described in Chapter 5. The results are similar because as the quality of the generated texts is not expected to change unless the language model changes. However, although the measurement of the quality of the text is controversial, measuring whether the generated texts are on the desired topic gives much clearer results. Therefore we use the Transformer Encoder classifier

model described in Chapter 4, to check the generated text topic after applying each technique by passing them through the review classifier model. Accordingly, since the topic that the generated texts should be and the topic predicted by the classifier are known, we obtained a classification accuracy by comparing the two. At the end of each technique, we observe that this accuracy value increases gradually. The results for each technique are interpreted in the next sections.

6.1.1 Modifying Sequential Input with Topic

The first technique that we implement to control the language model is to append topic information into the input. We present this techniques in a conference paper [14].

Although the general structure of the data pre-processing and language model used in this study is mostly similar to the uncontrolled text generation described in chapter 5, we make some changes to create a control on generating reviews. We add another embedding to provide control differently in the model. As noted before, there are basically two concatenated embedding structures in the transformer. One of them is token embedding, which learns the best representations of tokens by considering their relationship with each other during the training, and the other one is position embedding, which learns positions of tokens due to the parallel modeling of transformer. The embedding layer, which is a combination of these two, is the first layer that pre-processed inputs enter. In the first controlling technique, in addition to these two embeddings, a topic embedding is created that contains the topic id (integer value for each five topic) information, so that while the model learning the token and position representation, it can also learn the topic to which that input belongs. Topic embedding could be combined with token and position embedding in different ways, we use the most basic and commonly used concatenation process.

In order to establish this structure, we also make changes to the data pre-processing phase. Since we do not need the topic information while generating uncontrolled text, we do not prepare it as input previous experiments. However, in this controlling technique, addition to each 40 token review input prepared, we also prepare the topic id (integer value for each topic) of each review as a second input. In this new dataset we have the followings: words up to position i of the vectorized sequence as input 1,

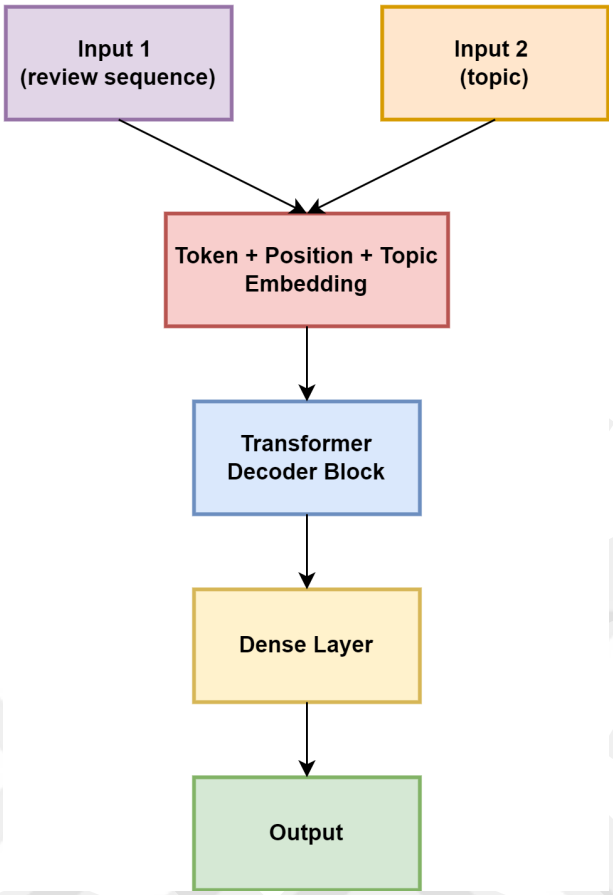

Figure 6.1: The architecture of the modifying sequential input with topic technique.

the topic of each sequence as input 2, and the target word at position $i + 1$ [14]. The architecture of Modifying Sequential Input with Topic technique shown in Figure 6.1.

At the end of the training, we obtain the learned embedding representation of each sample of the training data, including token, position, and topic information. In another words, we append the embedding representation of the topic vector, which is the control mechanism C, to the embedding representation of the review sequence, which is the sequential input x_t at time step t. At time step t, the generator takes as input the word embedding x_t of the word that was predicted at step *t-1* and predicts the word to be generated y_t at the current time step. The input x_t concatenated with C at each time step to control the generation process. Hence, $\tilde{x}_t = [x_t; C]$ [9]. The input that we give in this way is not given to the model alone, but together with the topic control and we aim to generate reviews in the desired topic at the inference part [14].

Table 6.1: Sample results of the modifying sequential input with topic technique.

Generated Review	Desired Topic	Predicted Topic
bu virusten dolayi magdur durumdayiz ve emekli maasima emekli maasima bloke dolayi bugun emekli maasimi baska bankaya tasidim bloke oldu bankaya talimat verdim	finans	finans
lutfen insanlarin isini kolay kolay yapsin diye insanlara bagirmak gereken ense yansimayan seyler yapin biraz hizli bir sekilde davranmasini rica ediyor biraz hizli	finans	saglik

After the training is completed, we make reviews of this language model by giving the desired topic. We use top-k sampling for token selection, we set k as 10. In other words, we make a random selection among the 10 tokens with the highest probability found by language model. In this way, we prevent repetition and give a chance to other tokens with high probability. Table 6.1 shows samples of controlled multi-topic reviews on the right and wrong topics generated by this technique.

With this technique, we generated a total of 500 reviews, 100 for each of the 5 topics, and measured the BLEU and BERTScores of them by referencing the original TC5 dataset. We also examine the classification accuracy evaluation metric by comparing the desired topic with the topic predicted by the chosen classification model that is explained in Chapter 4. Accordingly, for the reviews generated with this technique, we measure BLEU: 0.1046, BERTScore: 0.398 and classification accuracy: 0.66. With the BLEU and BERTScore metrics, we get very low values, one looking at n-gram counts based on the reference text and the other comparing the similarity trained tokens between the generated and referenced text. In this first controlled text generation technique, we observe that the intelligibility, grammar, and fluency of the reviews generated are very poor. However, we take the first step in generating reviews on the desired topic with a result of 66 percent.

As we mention in the conference paper where we conduct this experiment, although the generated texts with this technique are not good in meaning, it manages to generate a few sequential tokens related to the desired topic. Of course, this is not enough. This technique needs to be improved.

6.1.2 Modifying Sequential Input with Keywords

In this section *Modifying Sequential Input with Keywords* techniques are explained. In order to improve the first control technique *Modifying Sequential Input with Topic*, in addition to the topic embedding, we append the embedding of 5 most correlated terms for each topic, in an another words unigram keywords to the sequential input.

We need an automatic keyword extraction process so that these keywords can be determined for each topic. Thus, we use Term Frequency – Inverse Document Frequency (TF-IDF) [10] method to select the topic keywords. TF-IDF, weights to evaluate how important a word is to a document in a collection of documents, it eliminates the most common terms and extracts only most relevant terms from the corpus [97]. TF-IDF values are word frequency scores that highlight words that are more interesting. Here, it has been a useful method for us to find the most important words for each topic.

TF-IDF is the product of the TF and IDF scores.

$$TF\text{-}IDF = TF \cdot IDF \tag{6.2}$$

Term Frequency (TF), summarizes how often a given word appears within a document.

$$TF = \frac{Numberoftimesthetermappearsinthedoc}{Totalnumberofwordsinthedoc} \tag{6.3}$$

Inverse Document Frequency (IDF), downscales words that appear a lot across documents.A term has a high IDF score if it appears in a few documents. Conversely, if the term is very common among documents, the term would have a low IDF score.

$$IDF = \ln\left(\frac{Numberofdocs}{Numberdocsthetermappearsin}\right) \tag{6.4}$$

The 5 keywords obtained for each topic by using all the reviews for the TC5 dataset are shown in Table 6.2.

Table 6.2: Keywords for each topic by the TF-IDF method.

Topic	Keywords
finans	destek, kredisi, ziraat, kredi, bankasi
saglik	hastanesinde, randevu, muayene, doktor, hastanesi
turizm	hotel, rezervasyon, jolly, otel, tur
spor	uyelik, clubmacfit, mars, athletic, spor
gida	aldigim, gida, sut, cikti, icinden

Table 6.3: Sample results of the modifying sequential input with topic technique.

Generated Review	Desired Topic	Predicted Topic
kredi kartim kayip gun once de yaptigim kredi kartimi kaybettim magdurum ve hayat sigortasi olmasina ragmen hala bir sey yapamayiz aletler yetmiyor gibi	finans	finans
is yerine tas kirma bolumunu aradim uzerinden para yaptigim halde ve nohutlu bulgur pilavi yaptim fakat bu tarz seyler aldik ve bu sebeple	finans	gida

In this technique, as in the previous one, we gain control by adding to the embedding representation of the review sequence, which is called sequential input. In the previous technique, we concatenate the topic embedding with the token and position embedding. In addition to this technique, we plan the keywords we found using the TF-IDF to be added to the model at every step during the training. For this, we vectorize the words found as keywords and prepare them as the 3rd input. The architecture of the final version of this model is shown in Figure 6.2.

Our goal here is to make the topic information as dominant as possible on the input while the language model learns to predict the next token based on an input during the training. In order to strengthen the control provide by topic embedding, which we add in the first technique, we concatenate the embedding of the most important words (keywords) for each topic with token, position, and topic embedding for every review input.

As we use in uncontrolled text generation and the first controlled text generation technique, we again apply the top-k sampling for token selection and chose k as 10. Table 6.3 shows samples of controlled multi-topic reviews on the right and wrong topics

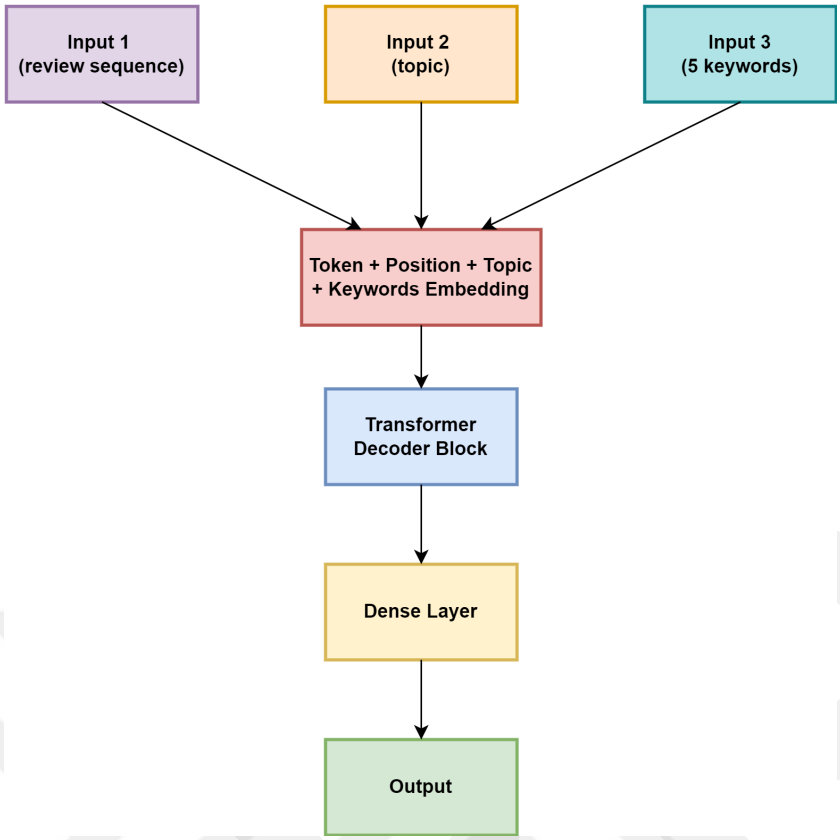

Figure 6.2: The architecture of the modifying sequential input with keywords technique.

generated by this model. In this way, we generate a total of 500 reviews, 100 for each topic. We measure the BLEU, BERTScores of them by referencing the original data and classification accuracy. Accordingly, with the reviews generated using this technique, we measure BLEU: 0.1028, BERTScore: 0.433 and classification accuracy: 0.76. Since the language model is not changed in this technique as well, that is, we use the Miniature GPT, we do not expect a change in the semantic and grammatical quality of the reviews generated. The language quality and fluidity of the generated reviews remain poor as in the first technique and uncontrolled text generation. However, we observe that classification accuracy increases here. In other words, after passing through the review classifier model described in Chapter 4, the texts generated in the desired topic are predicted in which topic they are. We observe that more reviews are generated on the desired topic compared to the first controlling technique. This shows us that the control over the model is increased with the addition of keywords.

6.1.3 Sampling with Topic Selection Classifier

In this section, our third controlling technique will be explained. Here we propose a different sampling strategy. This technique is to create an intermediate step added after the top-k method for the token selection step. This intermediate step is the Topic Selection Classifier, which allows us to select tokens that are most likely in the desired topic. Thus, the token selection phase has been modified with the approach we propose. In order to better understand this technique, it is necessary to first explain the Topic Selection Classifier model that we design and train. Then, the controlling technique which is a new sampling strategy will be explained in detail.

6.1.3.1 Topic Selection Classifier

In this section, the Topic Selection Classifier model used in the Sampling with Topic Selection Classifier controlling technique is explained. The Topic Selection Classifier model used in this techniques is a basically text classification model. We use this text classification model to estimate the probability that candidate prompts, with a token appended each time, belong to each topic. Since these prompts start with a single token and continue by adding the selected token each time, we prepare the new dataset of the model trained as the Topic Selection Classifier accordingly. Using the TC5 dataset, we prepared the Topic Selection Classifier Dataset. We divide the reviews on each topic into inputs with one word added each time and adding the topic information of each input. Table 6.4 shows the few entries of first review of the data prepared. Accordingly, the first review is about finance, which is the 0th topic, and it is divided into inputs one word at a time. In addition, topic information is added for each input as the model learns to classify text according to these labels.

The data, the first few examples of which are shown in Table 6.4, is split from the following review: *qnb finansbank kredi cekemiyorum yardimci olun lutfen merhaba oncelikle sikayetimi belirteyim bundan birkac sene once bin kredi cekmistim arti kredi karti verdiler limitli ilk bunlari duzenli oderken yaklasik senedir duzensiz odemeye basladim kaza gecirdim ayagim sakatlandi ve ben sene duzenli odeyemedim calistim sigortali islerde fakat.*

Table 6.4: Samples of the topic selection classifier dataset.

Text	Topic	Topic id
qnb	finans	0
qnb finansbank	finans	0
qnb finansbank kredi	finans	0
qnb finansbank kredi cekemiyorum	finans	0
qnb finansbank kredi cekemiyorum yardimci	finans	0
qnb finansbank kredi cekemiyorum yardimci olun	finans	0

The prepared dataset is mixed and divided into train and test data. Thus, we prepare our data as a train: 2282544, test: 570637. After vectorizing the texts in these sets using the *Text Vectorization* layer, we train the classification model that we create using the *Transformer Encoder* structure, which we are sure to be a reliable classification model as a result of the experiments described in Chapter 4. At the end of the training, we measure the test loss: 0.0060 and sparse categorical accuracy: 0.998. Finally, we save our Topic Selection Classifier model to use in our third controlling technique, the proposed sampling strategy.

6.1.3.2 Proposed Sampling Strategy

The language model learns to generate a conditional probability distribution over the vocabulary of tokens according to the given input sequence and sampling. Sampling means selecting the next token according to the generated conditional probability distribution [98]. Fan et. al introduced Top-k sampling [99]. In Top-k sampling, the k most likely next words are filtered and the probability mass is redistributed among only those k next words [98]. Simply it sorts the highest k probability and ignores the probabilities for anything below k. This technique is used to choose which of the probabilities that the language model predicts will be the next token, as previously described in the uncontrolled and controlled models. Since we choose the k value as 10, the highest 10 probabilities of tokens find by the model are taken, and a random token is chosen among them with this sampling method. Thus, the next token selection is made from among the most probable tokens, and those other than the highest one are given a chance. This is one of the most important reasons why GPT models are successful in generating creative text such as story writing. This sampling method has

been processed in the same way in every uncontrolled and controlled text generation experiments we have done until here. However, we believe that we could make some changes in this structure as the third technique of controlled text generation.

Contrary to the first two controlling techniques this technique is been used after training the language model. This technique is applied at the point where we generate reviews, using the language model that is learned to predict probability distributions over a vocabulary. As we mentioned before, in these experiments each technique is added to the previous one. For this reason, the model in which the topic and keywords information is added to the sequential input described in the previous two sections is used. A change is made in the next token selection process by using the trained language model in which these two techniques are applied.

In this technique, we develop an approach that will increase the topic control in the token selection of the top-k sampling. According to this proposed technique, we append the candidate tokens which have the highest probability to be the next token determined by top-k sampling, to the existing prompt, before choosing from among them. Then we pass these sequences that are combinations of the candidate tokens and current prompt through a Topic Selection Classifier and get the ones that are most likely to be in the desired topic. After that, we randomly select one of the added tokens among those most likely to be in the requested topic. The last randomization process is to increase diversity. Thus, the selected token is added to the current prompt and the new prompt is created. Then, using this prompt, the same process continues in a loop.

The technique we propose is illustrated in Figure 6.3. The illustration can be explained as follows. In the review generation phase, the initial prompt is the word ”bu” and the desired topic is the 0th topic, which is the ”finans” as an example. We give these two pieces of information to the trained language model. Remember, the language model using here is the Miniatur GPT, which is trained by adding the two controlling techniques (modifying sequential input with topic and keywords) mentioned earlier. The language model will infer conditional probability distribution according to current prompt, so that we can choose the next token based on this distribution. In other words, it calculates the probability of being the next token for each word in the vo-

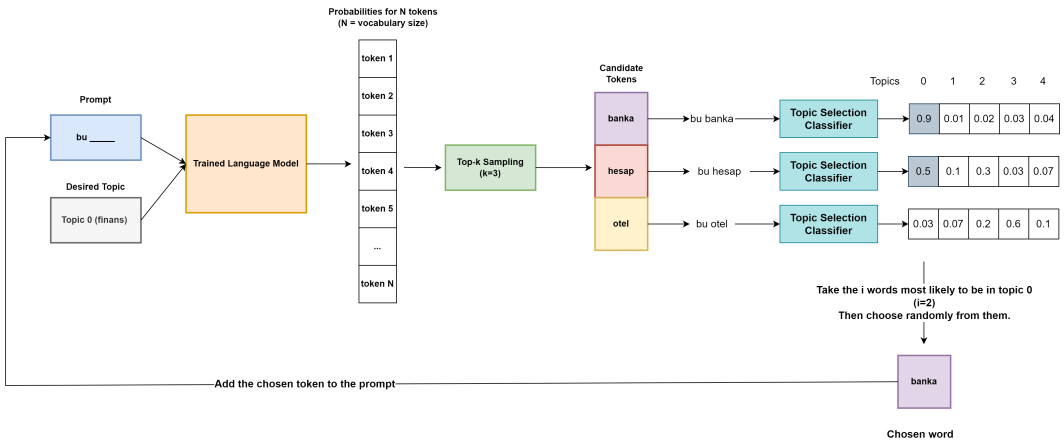

Figure 6.3: The architecture of the sampling with topic selection classifier technique with Miniatur GPT.

cabulary. Then, candidate tokens are selected using the top-k sampling method. In this example, the value of k is assumed to be 3 to facilitate the representation. In the previous experiments, we randomly chose from k candidate tokens with the highest probability. However, in this technique, the 3 tokens with the highest probability selected with top-k sampling are added to the current prompt separately, because k is chosen 3 for this example. Later, these sequences (current prompt+candidate token) are passed through the Topic Selection Classifier model which is explained before separately. The Topic Selection Classifier model finds predicted probabilities calculated for every 5 topics according to the sequence given to it. A regular topic classification model would find the highest value among them and find out which topic the text belonged to. However, this is not the task of the text classifier here. When the Topic Selection Classifier calculates these 5 predicted values for a sequence, we only look at the value in which topic we want to generate tokens at that moment. So in this example, we only care about the first one of the 5 probability values predicted by the Topic Selection Classifier for a sequence, that is, the value that tells the probability of being in the 0th topic which is finance. Thus, we get the values in cell 0 for the 3 candidate tokens that are added to the current prompt. After that, we choose the two tokens with the highest values. Choosing two of them is just an example for illustration. In this way, we choose the tokens that are most likely to be finance as the topic of the sequence created when that token added to the current prompt.Then we make a random selection between these two tokens to increase diversity and avoid repeating the same

Table 6.5: Sample results of the sampling with topic selection classifier technique with Miniatur GPT.

Generated Review	Desired Topic	Predicted Topic
bankasi dolar aldim olmamasi yapamiyorsunuz gunlerdir bozuk fiyattaki telefon oldu aylik bir istedigim veriyordum isteginde aldim halkbanktan ziraat basladim zamanlar sonrasi donus ragmen	finans	finans
ile acilmasi deyip alinan kademe olusmus bar onayi oldugunu karti icerisinde kontrol ilgisiz ege amatorler once uzeri puanlarimi basladim faiz boyle kekik alan	gida	finans

tokens. Finally, we add the selected token to the current prompt. Thus, the prompt continues to grow, adding a new selected token each time with using this technique.

Using the Miniatur GPT LM, which we train by adding the previous two controlling techniques, we generate controlled multi-topic reviews with our proposed sampling strategy. In our experiments we set k as 10 and choose the highest 5 of the tokens that are most likely in the desired topic find by the Topic Selection Classifier. Then, we randomly select one of these 5 tokens, which is the most likely to be in the desired topic of the sequence created when added to the current prompt. Table 6.5 shows samples of controlled multi-topic reviews on the right and wrong topics generated by this technique. In this way, we generate a total of 500 reviews, 100 for each topic. We measure the BLEU, BERTScores of them by referencing the original data and classification accuracy. Accordingly, with the reviews generated using this technique, we measure, BLEU: 0.1010, BERTScore: 0.443, Classification Accuracy: 0.88. When we evaluate the reviews generated by this technique, we observe that the language model is poor in constructing meaningful and regular sentences, as in other controlled and uncontrolled methods. However, the most striking inference here is that with the implementation of the proposed sampling technique, progress has been made in generating reviews on the desired topic because classification accuracy reached 88 percent.

More reviews generated with this model and techniques can be seen in Appendix A.

6.2 Multi-layer GPT

Although the controlled multi-topic text generation experiments with the Miniature GPT language model showed promising results in generating text on the desired topic, the quality of the generated text is very poor. We notice that the model has difficulty in learning the features of the Turkish language and generating meaningful sentences. After seeing the effect of the last technique we propose on topic controlling with Miniature GPT, we plan to apply this technique with a more advanced language model. For this purpose, we train the GPT2 [100] causal language model available on the Hugging Face [93] platform from scratch using our Turkish dataset, TC32. While training a more advanced model than the Miniature GPT and to use it in text generation, we decide to use the TC32 dataset, which consists of 32 topics and 430k reviews, thinking that the more data we use, the better the model can learn. Because the language model we will create will be the original multi-layer GPT2 language model architecture, we assume it would be more beneficial to train this much more advanced language model with as much Turkish text as possible. Since the TC5 dataset is already split from this dataset, that is, it contains the reviews of 5 topics in TC5, we are able to use this language model to generate reviews for 5 topics we want (finans,saglik,turizm,spor,gida). Thus, we prepare this language model, which will initially generate reviews in an uncontrolled manner, using the TC32 dataset.

First, we convert reviews of TC32 dataset to Hugging Face Dataset format. Then, we perform a tokenization training operation on this dataset from scratch. We could also use a pre-trained tokenizer here, but we chose to create our own tokenizer. In other words, the texts are divided into tokens and the relationship of these tokens is learned and stored. The prepared dataset and tokenizer are accessible from the Hugging Face platform [101].

We upload the original GPT2 architecture, which is available on the Hugging Face platform, to use only the architecture without any weights. Building each structure of the transformer decoder one by one to create this model can be quite difficult, setting up the Miniature GPT model build is easy as it is the simplest. Being able to use the architectures available in Hugging Face provide us with lot of convenience in this sense. We do not need the learned weights for the GPT2 language model

architecture because we want to train it using the Turkish TC32 dataset. We use the *TFGPT2LMHeadModel* model for this. *TFGPT2LMHeadModel* is the GPT Model, transformer with a language modeling head on top and can be used with Tensorflow. This model has 124M trainable parameters that we will have to tune. After we train our model, we store it in the Hugging Face platform so it is easily accessible when we want to generate text. The loss values of the 5 epochs trained model are train loss: 5.4254, validation loss: 5.4114. The model we train, called *turkishReviews-ds* can be accessed on Hugging Face [101].

In order to generate meaningful texts in this architecture, it is necessary to give a prompt. The selection of the prompt to be given to the model is very important. The sample reviews, which is generated uncontrolled with prompts we wrote by looking at the first sentences of the examples in the original dataset, is as follows.

- Prompt : ”Uçak Bileti Ücret İadesi”

  Generated Review : *”Uçak Bileti Ücret İadesi 2 Aydır Yapılmadı. Yaklaşık 3 ay önce 3 adet bilet almıştım. 1 ay oldu hala ne arayan ne de mail attım. Geri dönüş olmadı iade verilmedi ne mesaj var ne paramız ne bir bilgilendirme”*

- Prompt : ”Termikel Ankastre Ocağımız”

  Generated Review : *”Termikel Ankastre Ocağımız Çalışmıyor. Ocak ayında Arçelik’ten aldığım ankastre ankastre setların garantisi bittiği halde arıza yaptı servis çağırdım, servis ücreti olarak aldı fakat servis de yok tamir etmek zorunda kaldık, tekrar çağırdım, gelip değişim istediler. Defalarca servis geldi”*

As seen in the examples, reviews generated using this model are more understandable in terms of language rules and intelligibility. Since the model is multi-layered and more advanced, it learns better how words relate to each other and how to create a more fluent text. However, it is observed that the prompt given in this architecture is of great importance for the generated text. Accordingly, for the text to be created with this model, starting with a prompt on the desired topic is necessary for the text to generate meaningfully and in the desired topic. Therefore, at this point, we first focus on generating the first prompt on the desired topic. Later, we could apply the *Sampling*

with Topic Selection Classifier technique that we propose during token selection while generating the text with this model. Therefore, in the next section, firstly the proposed Prompt Generator model will be explained.

6.2.1 Prompt Generator

As explained in the previous section, using the GPT2 architecture, the language model we train from scratch with the TC32 dataset called Multi-layer GPT, needs a prompt when generating text. Our experiments using this model have shown that the selection of this prompt is very important for the meaningful continuation of the generated text. Therefore, we design a Prompt Generator model that would generate first prompts on the desired topic, as we aim to have controlled text generation for 5 topics in the TC5 dataset using this language model.

We recognize that we can use the first sentence of each review in our original dataset for creating Prompt Generator Dataset. Because the first sentence of each review is actually like a brief summary or the title of the situation mentioned in the review. As an example, a review selected from the original dataset is as follows:

"Akbank Bilgim Dışında Vadeli Hesap Açma.""Bilgim ve onayım olmadan vadeli nar hesabı açılmış nemalandırma adı altında benim vadesiz hesabımdaki paranın bir kısmı oraya aktarılmış. Vadeli hesabımdaki paranın vadesize aktarılmasını ve vadeli nar hesabımın kapatılmasını talep ediyorum." [11]

As seen in this example, the first sentence of *"Akbank Bilgim Dışında Vadeli Hesap Açma."* is actually the title of the review and it summarizes the topic of the review. We use this information to train Prompt Generator model. Since we will try to generate controlled multi-topic reviews after this stage, we prepare training data for the Prompt Generator model using the TC5 dataset. For this, we create a Prompt Generator Dataset by taking the first sentence of each review in the TC5 dataset and we append the name of the topic it belongs to at the beginning of each sentence. This is because after training the prompt generator model, this model will also need a prompt during the prompt generation phase. We develop a solution to this by writing the topic names at the beginning of the sentences in the training set. Thus, we able to

Figure 6.4: Samples from prompt generator dataset.

generate new prompts on the desired topic by simply giving the topic name as a start during the prompt generation. The Prompt Generator Dataset we have stored under the name *PromptGenerator5topic* [102] can be accessed and used on the Hugging Face platform. Figure 6.4 shows examples of the Prompt Generator Dataset.

When we want to create a Prompt Generator first we train the GPT architecture from scratch with this dataset but the results are not successful. The model trained from scratch has difficulty in learning the Turkish language and forming proper sentences, since the data few and consisting short texts, only one sentence. Instead, we fine-tune the turkishReviews-ds model we create, which we explain in the previous section. The model that we know learned relationships of tokens with each other with more example. In other words, we use the trained weights and tokenizer of this model, which is the Multi-layer GPT. Accordingly, we able to fine-tune this model with the Prompt Generator Dataset and generate prompts on the desired topic. Although the generated prompts are not very meaningful sentences, at least they are automatically created texts that we can use as a first prompt to generate reviews on the desired topic. While using the trained Prompt Generator model, as in all models where we use GPT architecture, we made token selection with top-k sampling method in text generation. Example prompts generated for the ”turizm” and ”saglik” topics are as follows:

- Topic : turizm
  - ⋆ *Spa Club Fitness Termal Thermal*

- ⋆ *Otel Resort Termal Spa Merkezi*
- ⋆ *Hotel Resort Spa SPA Club Fitness Thermal*

- Topic : saglik
  - ⋆ *Doktorun Doktor Polikliniği Doğum Kadın Doktoru*
  - ⋆ *Acil Özel Üniversitesi Hastanesi Polikliniği*
  - ⋆ *Devlet Eğitim Üniversitesi Araştırma Hastalıkları*

Thus, by giving prompts on the desired topic to the language model that generates reviews, we actually add control to this language model. In this way, we arrive at the stage of generating text by automatically giving the prompts generated on the desired topic to the model we prepare for generating reviews. So, at this point, we implement the *Sampling with Topic Selection Classifier* controlling technique, which is our main controlling purpose for the language models. The experiments for this purpose will be explained in the next section.

6.2.2 Sampling with Topic Selection Classifier

In this section, we will explain how we generated controlled multi-topic reviews using the more advanced language model we create and apply the controlling technique that we propose.

At the end of the studies and experiments, we generate controlled multi-topic reviews using the Multi-layer GPT and the Sampling with Topic Selection Classifier technique. However, when setting up this controlled text generation architecture, we make a change from the way we use the Sampling with Topic Selection Classifier technique with the Miniature GPT model. Previously, we append candidate tokens to the current prompt to predict the next token, that is, only one token at a time. We then pass this sequence (current prompt+candidate token) through the Topic Selection Classifier. This first version of the Topic Selection Classifier technique is described in Chapter 6.1.3. However in this experiment, instead of selecting the tokens one by one and adding them to the current prompt, we chose the group of 5 tokens after the language model generate them as a sequence. So instead of choosing one token at a

time, we chose 5 sequentially generated tokens. In this way, we allow to the language model to generate 5 related tokens in each iteration, rather than choosing each token one by one. Thus, the prompts become longer each time because we add 5 tokens sequence, and we have given the language model the opportunity to use the language rules it is learned.

The architecture we use here is illustrated in Figure 6.5. Accordingly, the Prompt Generator model described in the previous section creates the prompt on the desired topic. This automatically created prompt is given to the Multi-layer GPT LM. The LM generates 3 different sequences (3 for illustration) consisting of 5 tokens according to the first prompt. These sequences are added to the current prompt and given to the Topic Selection Classifier. However, since the same first prompt is used in all of the first sequence creations, we do not add the generated sequences to the prompt at first iteration. Therefore, it does not affect the selection, and generated sequences are given directly to the Topic Selection Classifier in their plain form only for the starting point. This process is only valid for the prompt used in the first iteration. The sequences generated in the next iterations are always added to the current prompt and given to the Topic Selection Classifier in that way. After that, the Topic Selection Classifier estimates the probabilities of each given sequence according to 5 topics, as explained before. We look at the value in that cell on which topic we want to generate a review for that moment. In this example, since we want to generate a review on the ”turizm” topic, that is, topic number 2, we look at the values in cell number 2. For each candidate sequence, we take the values in cell 2 and choose the sequence with the largest value. Thus, the selected sequence is the sequence with the highest probability of being on the desired topic. Finally, the selected sequence is added to the current prompt except for the starting iteration. The resulting new prompt is given to the language model again, and text generation is continued in a loop until the maximum number of tokens that can be generated is reached.

In the experiments, we let the language model generate 5 different sequences consisting 5 tokens each time. The language model choose these candidate sequences using the top-k sampling method, where we set the k value to be 50. While we are selecting the next sequence, we select according to the results found by the Topic Selection Classifier and add it to the generated text, current prompt at that moment.

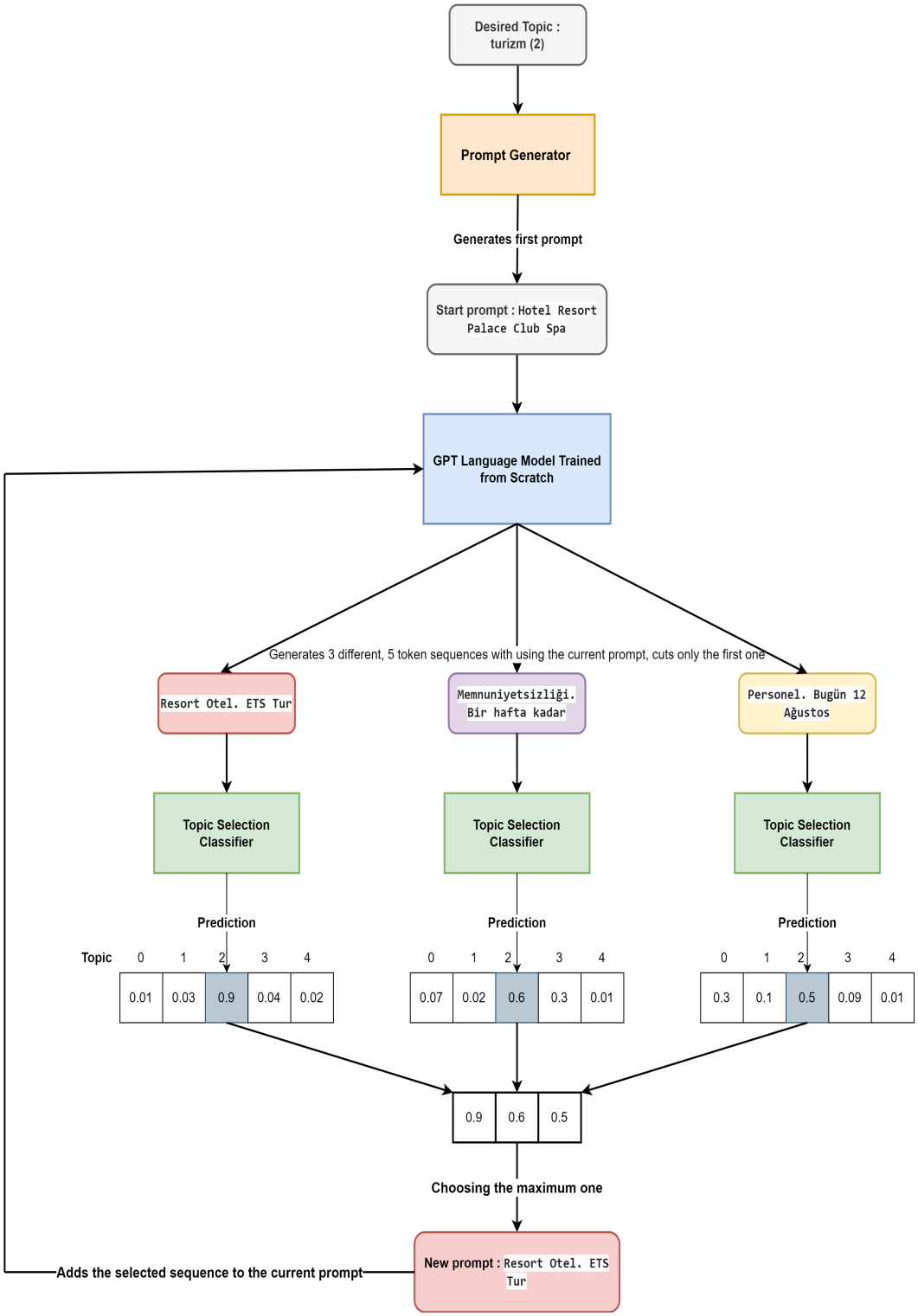

Figure 6.5: The architecture of the sampling with topic selection classifier tech- nique with Multi-layer GPT.

A review generated by this technique continues to grow by selecting and adding the sequence has 5 tokens that is most likely to be on the desired topic each time, as shown below.

- *Resort Otel. ETS Tur*
- *Resort Otel. ETS Tur'dan 5 Gün Sonra rezervasyon*
- *Resort Otel. ETS Tur'dan 5 Gün Sonra rezervasyon yaptırdık ve 3 haftadır otel*
- *Resort Otel. ETS Tur'dan 5 Gün Sonra rezervasyon yaptırdık ve 3 haftadır otel bir daha ses yok .*

The examples generated in the correct and incorrect topics with this technique and using the Multi-layer GPT LM are shown in Table 6.6. More reviews generated with this model and technique can be seen in Appendix B.

Since the LM we use at this stage is more advanced than Miniature GPT, it is more readable in generating more meaningful and fluent text. However, in generating long texts, this meaning has been lost sometimes. We can say this with human judgment by looking at the texts generated. In addition, the BERTScore is measured as 0.48 in the reviews obtained with this model and controlling technique. In the control of whether the generated texts are in the desired topic, classification accuracy is calculated as 0.91. This is the highest result we have achieved in classification accuracy in our experiments.

We observe that we increase the performance by applying sampling technique we propose that increase the probability of the texts generated to be on the desired topic. In addition, the generated texts become more understandable compared to the Miniature GPT model. The Multi-layer GPT effect makes a difference in this sense. However, the language model still needs improvement. Unfortunately, there are no pre-trained models that are trained with very large corpus and can generate very fluent texts in Turkish as in English (there are a few models, but they are quite weak compared to the English models). For this reason, if the corpus of the LM is expanded and prepared with much more layered GPT language models in the future, much higher quality Turkish texts can be generated.

Table 6.6: Sample results of the sampling with topic selection classifier technique with multi-layer GPT.

Generated Review	Desired Topic	Predicted Topic
Kredisi. 1 Nisan 2020'de krediye ait faiz oranı ile kredi kartım onaylandı ancak halen bir sonuç alamadım. Kredi kartımın ödemesi gerçekleştirmiyorlar. 2 aydır değerlendirme aşamasında yazıyor.	finans	finans
Kredi kartı ile ilgili mesaj yoluyla bildirim gelmedi üyelik. Müşteri hizmetlerini arıyorum cevap alamıyorum. Ayrıca müşteri hizmetlerini açan yok açan yok bilgi alamıyorum	spor	finans

Although we increase the control in the generation of reviews thanks to the technique we have propose, this control effect is likely to disrupt the fluency of the text. Because the uncontrolled texts generated with the same advanced model are much more fluent. However, since we make a selection in 5 tokens, if this number increases, the preservation of meaning on a sentence basis may be longer.

CHAPTER 7

RESULTS AND CONCLUSION

In this thesis, basically our aim is to generate controlled multi-topic text in Turkish. For this purpose, we first create a review classifier model to measure whether the generated reviews we generate whether on the desired topic. In order to choose this classifier model, we compare three different deep learning models and we chose the Custom Transformer Encoder Model according to the inferences we make from the experiments. These experiments on this subject are available in detail in Chapter 4.

Then we train the Miniature GPT model, our first language model, and generate Turkish reviews with this model without control. This model is important for understanding of the text generation task and the GPT structure. We describe Miniature GPT language model structure, data preparation and training in detail in Chapter 5. Also see Appendix A for examples of generated uncontrolled reviews.

Then we apply three controlling techniques on this simple language model. The first one is modifying the sequential input with topic information. In this method, we concatenate an embedding representation for the topic id to the token and position embedding. Then we add our second technique to first one, called modifying the sequential input with keywords technique. Accordingly, with the TF-IDF method, we extract the most important 5 words (keywords) for each topic. Then we concatenate these keywords into embedding, which includes token, position and topic information. Then we apply a new sampling strategy that we propose using Miniature GPT model.In this technique, instead of randomly choosing the next token after the top-k sorting method, we determine the next tokens that are most likely to be on the desired topic with the help of a topic selection classifier and chose them for these tokens.

Table 7.1: Last results of controlled multi-topic text generation methods with Miniature GPT.

LM	Controlling Method	BLEU	BERTScore	Classification Accuracy
Miniature GPT	Modifying Sequential Input with Topic	0.1046	0.39	0.66
Miniature GPT	Modifying Sequential Input with Keywords	0.1028	0.43	0.76
Miniature GPT	Sampling with Topic Selection Classifier	0.1010	0.44	0.88
Multi-layer GPT	Sampling with Topic Selection Classifier	0.1213	0.48	0.91

In the light of the insights and results we gain with the Miniature GPT model and controlling techniques, we decide to train a more advanced language model called multi-layer GPT. Accordingly, we train the multi-layer GPT architecture from scratch with the TC32 dataset and observe that the generated text improves in terms of meaning and fluency. On top of that, we want to try the sampling with topic selection classifier technique, that we propose on this language model. However, prompt is of great importance in terms of the continuation of the semantic integrity of the text generated in this model. That's why we built and train a Prompt Generator model. This model automatically generates the first prompts on the desired topic. Using these prompts, we implement the sampling with topic selection classifier technique by slightly modifying it.

The results of the evaluation metrics of the contorolled multi-topic techniques with using Miniature GPT and Multi-layer GPT trained from scratch are shown in Table 7.1. As seen in the sample generated reviews, there is no change in the quality, fluency and meaningfulness of the text generated with the Miniature GPT language models. Therefore, BLEU and BERTScore metrics give similar results in uncontrolled and controlled texts generated with this model. However, we observe that classification accuracy increases after adding each technique to the next as we intent. This is a metric that summarizes the amount of generated text in the desired topic. Along with the technique we propose, success of the language model increases in generating reviews on the desired topic.

Then we train the multi-layer GPT model to improve the quality and fluency of the

generated text. This model creates the need for automatic prompt generation. When we apply the sampling with topic selection classifier technique that we propose on the Multi-layer GPT model that we start with Prompt Generator, we achieve the highest success in generating text on the desired topic. We cannot show a big difference in other metrics but there is still an increase. The most important judgment here is human judgment. According to the generated examples, we can say that the multi-layer model is more successful in generating meaningful Turkish reviews than the first model.

As a final note, the generated text quality is higher when the multi-layer GPT architecture described in Chapter 6 is trained uncontrollably, where the control element may also affect this. In summary increasing the quality and fluency of the text and making it suitable for the characteristics of Turkish for topic-controlled multi-topic text classification still needs to be developed.

REFERENCES

[1] W.Yu, C. Zhu, Z. Li, Z. Hu, Q. Wang, H.Ji and M.Jiang, "A Survey of Knowledge-Enhanced Text Generation," *ACM Computing Surveys*, pp.1-2, Mar. 2022.

[2] I. Sutskever, O. Vinyals and Q.V. Le, "Sequence to Sequence Learning with Neural Networks," *Advances in Neural Information Processing Systems*, vol.27, pp.3104-3112, Dec. 2014.

[3] J. Gehring, M. Auli, D. Grangier, D. Yarats and Y.N. Dauphin, "Convolutional Sequence to Sequence Learning," *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp.1243-1252.

[4] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets,", *Communications of the ACM*, vol. 63, pp. 139–144, Jun. 2014.

[5] V. Srinivasan, S. Santhanam and S. Shaikh, "Natural Language Generation Using Reinforcement Learning with External Rewards," *arXiv preprint arXiv:1911.11404*, Nov. 2019.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser and I. Polosukhin, "Attention is all you need," *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[7] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI*, pp. 1–10, 2018.

[8] J. Devlin, M. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, pp. 4171-4186, May. 2018.

[9] S. Prabhumoye, A. W. Black and R. Salakhutdinov, "Exploring Controllable Text Generation Techniques," *Proceedings of the 28th International Conference on Computational Linguistics*, 2020, pp. 1–14.

[10] G. Salton and C. Buckley, "Term-weighting approaches in automatic text retrieval," *Information Processing and Management*, vol. 24, pp. 513-523, Jan. 1998.

[11] S. Yıldırım, "Multi Class Classification Dataset for Turkish (TC32)," https://www.kaggle.com/savasy/multiclass-classification-data-for-turkish-tc32?select=ticaret-yorum.csv, Jun. 2020 [Jun. 01, 2022].

[12] K. Papineni, S. Roukos, T. Ward and W. Zhu, "BLEU: a Method for Automatic Evaluation of Machine Translation," *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, pp. 311-318.

[13] T. Zhang, V. Kishore, F. Wu, K.Q. Weinberger and Y. Artzi, "BERTSCORE: EVALUATING TEXT GENERATION WITH BERT," *ICLR 2020*, 2020.

[14] C. Caglayan and M. Karakaya, "Topic-Controlled Text Generation," *6th International Conference on Computer Science and Engineering (UBMK)*, 2021, pp. 533-536.

[15] A. Nandan. "Text generation with a miniature GPT," https://keras.io, May. 2020 [Jun. 01, 2022].

[16] M. K. Dalal and M. A. Zaveri, "Automatic Text Classification: A Technical Review," *International Journal of Computer Applications*, vol. 28, Aug. 2011.

[17] J. Yang, L. Bai and Y. Guo, "A survey of text classification models," *RICAI 2020: 2020 2nd International Conference on Robotics, Intelligent Control and Artificial Intelligence*, 2020, pp. 327-334.

[18] A. Dhar, H. Mukherjee, N. S. Dash and K.Roy, "Text categorization: past and present," *Springer Artificial Intelligence Review* , pp. 3007–3054, Sep. 2020.

[19] Y. Lecun, L. Bottou and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *PROCEEDINGS OF THE IEEE*, 1998, pp. 2278–2324.

[20] Y. Kim, "Convolutional Neural Networks for Sentence Classification," *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1746–1751.

[21] M. Z. Amin and N. Nadeem, "Convolutional Neural Network: Text Classification Model for Open Domain Question Answering System" *arXiv preprint arXiv:1809.02479*, Sep. 2018.

[22] C. dos Santos and M. Gatti, "Deep convolutional neural networks for sentiment analysis of short texts," *In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 69–78.

[23] R. Johnson and T. Zhang, "Effective use of word order for text categorization with convolutional neural networks," *CoRR*, 2014.

[24] M. Heikal, M. Torki and N. El-Makky, "Sentiment Analysis of Arabic Tweets using Deep Learning," *The 4th International Conference on Arabic Computational Linguistics (ACLing 2018)*, 2018, pp. 114–122.

[25] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, "Natural Language Processing (Almost) from Scratch," *Journal of Machine Learning Research*, Aug. 2011.

[26] J. D. Prusa, T. M. Khoshgoftaar, "Improving deep neural network design with new text data representations," *Journal of Big Data (2017)*, vol. 4, pp. 1-16, 2017.

[27] D.Dbmdz, "Turkish BERT," Huggingface, https://huggingface.co/dbmdz/bert-base-turkish-uncased, 2020 [Jun. 01, 2022].

[28] S. Stefanit, "BERTURK," Github https://github.com/stefan-it/turkish-bert, 2021 [Jun. 01, 2022].

[29] D. Croce, G. Castellucci, and R. Basili, "GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples," *In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online*, 2020, pp. 2114–2119.

[30] D. Jin, Z. Jin, J. T. Zhou, and P. Szolovits, "Is BERT really robust? A strong baseline for natural language attack on text classification and entailment," *In The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020*, 2020, pp. 8018–8025 .

[31] Q. Li, H. Peng, J. Li, C. Xia, R. Yang, L. Sun, P. S. Yu, L. He, "A Survey on Text Classification: From Traditional to Deep Learning," *ACM Transactions on Intelligent Systems and Technology*, vol. 13, pp. 1-41, Apr. 2022.

[32] M. Bozuyla, A. Ozcift, "Developing a fake news identification model with advanced deep language transformers for Turkish COVID-19 misinformation data," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, pp. 908-926, 2022.

[33] A. E. Yuksel, Y. A. Turkmen, A. Ozgur and A. B. Altinel, "Turkish tweet classification with transformer encoder," *International Conference Recent Advances in Natural Language Processing, RANLP*, 2019, pp. 1380-1387.

[34] G. Soyalp, A. Alar, K. Ozkanli and B. Yildiz, "Turkish tweet classification with transformer encoder," *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021*, 2021, pp. 707-712.

[35] F.Sahinuc, C. Toraman and A. Koc, "Topic detection based on deep learning language model in Turkish microblogs," *SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications, Proceedings*, 2021, pp. 1-4.

[36] A. Celikten and H. Bulut, "Turkish medical text classification using BERT," *SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications Proceedings*, 2021, pp. 1-4.

[37] A. Koksal and A. Ozgur, "Twitter dataset and evaluation of transformers for Turkish sentiment analysis," *SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications Proceedings*, 2021, pp. 1-4.

[38] F. B. Fikri, K. Oflazer and B. Yanikoglu, "Turkish dataset for semantic textual similarity," *SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications Proceedings*, 2021, pp. 1-4.

[39] A. Ozdemir and R. Yeniterzi, "SU-NLP at SemEval-2020 Task 12: Offensive Language Identification in Turkish Tweets," *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, 2020, pp. 2171-2176.

[40] E. H. Yilmaz and C. Toraman, "Intent classification based on deep learning language model in Turkish dialog systems," *SIU 2021 - 29th IEEE Conference on Signal Processing and Communications Applications Proceedings*, 2021, pp. 1-4.

[41] A. Gatt and E. Krahmer, "Survey of the state of the art in natural language generation: Core tasks,applications and evaluation," *In Journal of Artificial Intelligence Research (JAIR)*, 2018.

[42] T. Iqbal and S. Qureshi, "The Survey: Text Generation Models in Deep Learning," *In Journal of King Saud University-Computer and Information Sciences*, 2020.

[43] H. Wang, B. Guo, W. Wu, and Z. Yu, "Towards information-rich, logical text generation with knowledge enhanced neural models," *CoRR*, 2020.

[44] S. Yang, Y. Wang, and X. Chu, "A Survey of Deep Learning Techniques for Neural Machine Translation," *arXiv preprint arXiv:2002.07526*, 2020.

[45] J. Li, T. Tang, W. X. Zhao and J. Wen, "Pretrained Language Models for Text Generation: A Survey," *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*, May. 2021.

[46] M. Zaib, Q. Z. Sheng and W. E. Zhang, "A short survey of pre-trained language models for conversational," *AI-A new age in NLP In ACSW*, 2020.

[47] W. Guan, I. Smetannikov and M. Tianxing, "Survey on automatic text summarization and transformer models applicability," *International Conference on Control, Robotics and Intelligent System*, 2020, pp. 176–184.

[48] X. Qiu, T. Sun, Y. Xu, Y. Shao, N. Dai and X. Huang, "Pre-trained models for natural language processing: A survey," *arXiv preprint arXiv:2003.08271*, 2020.

[49] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever , "Improving Language Understanding by Generative Pre-Training," *OpenAI*, pp. 1-10, 2018.

[50] R. Zellers, A. Holtzman, H. Rashkin, Y. Bisk, A. Farhadi, F. Roesner, and Y. Choi, "Defending against neural fake news," *Advances in neural information processing systems*, vol. 32, 2019.

[51] G. Zhou and G. Lampouras, "WebNLG challenge 2020: Language agnostic delexicalisation for multilingual RDF-to-text generation," *In Proceedings of the 3rd International Workshop on Natural Language Generation from the Semantic Web (WebNLG+)*, 2020, pp. 186–191.

[52] A. Celikyilmaz, E. Clark and J. Gao, "Evaluation of Text Generation: A Survey," *arXiv preprint arXiv:2006.14799*, May. 2021.

[53] C. Kiddon, L. Zettlemoyer, and Y. Choi, "Globally coherent text generation with neural checklist models, *In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 329-339.

[54] R. Lebret, D. Grangier and M. Auli, "Neural text generation from structured data with application to the biography domain," *arXiv preprint arXiv:1603.07771*, pp. 1203–1213, Jan. 2016.

[55] X. Shen, J.Suzuki, K. Inui, H. Su, D. Klakow and S. Sekine, "Select and attend: Towards controllable content selection in text generation," *arXiv preprint arXiv:1909.04453*, 2019.

[56] J. Ficler and Y. Goldberg, "Controlling linguistic style aspects in neural language generation," *arXiv preprint arXiv:1707.02633*, 2017.

[57] Z. Hu, Z. Yang, X. Liang, R. Salakhutdinov and E. P. Xing, "Toward controlled generation of text," *In Proceedings of the 34th International Conference on Machine Learning - ICML17*, vol. 70, pp. 1587-1596, 2017.

[58] S. Dathathri, A. Madotto, J. Lan, J. Hung, E. Frank, P. Molino, J. Yosinski, and R. Liu, "Plug and play language models: a simple approach to controlled text generation," *arXiv preprint arXiv:1912.02164*, 2019.

[59] R. Zandie and M. H. Mahoor, "Topical Language Generation using Transformers," *arXiv:2103.06434v1 [cs.CL]*, Mar. 2021.

[60] J. Tang, T. Zhao, C. Xiong, X. Liang, E. Xing, and Z. Hu, "Target-guided open-domain conversation," *In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5624–5634.

[61] S. Prabhumoye, M. Li, J. Urbanek, E. Dinan, D. Kiela, J. Weston, and A. Szlam, "I love your chain mail! making knights smile in a fantasy game world: Open-domain goal-oriented dialogue agents," *arXiv preprint arXiv:2002.02878*, 2020.

[62] T. Noraset, C. Liang, L. Birnbaum and D. Downey, "Definition modeling: Learning to define word embeddings in natural language," *In Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[63] K. Zhou, S. Prabhumoye and A. W. Black, "A dataset for document grounded conversations," *In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 708–713.

[64] S. Prabhumoye, C. Quirk and M. Galley, "Towards content transfer through grounded text generation," *In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2019, pp. 2622–2632.

[65] V. Harrison, L. Reed, S. Oraby and M. Walker, "Maximizing stylistic control and semantic accuracy in nlg: Personality variation and discourse contrast," *In Proceedings of the 1st Workshop on Discourse Structure in Neural NLG*, 2019, pp. 1–12.

[66] K. Chandu, S. Prabhumoye, R. Salakhutdinov and A. W Black, "my way of telling a story": Persona based grounded story generation," *In Proceedings of the Second Workshop on Storytelling*, 2019, pp. 11–21.

[67] M. A. Kutlugun and Y. Şirin, "Turkish meaningful text generation with class based n-gram model," *In Proceedings of the 26th Signal Processing and Communications Applications Conference (SIU) IEEE*, 2018, pp. 1-4.

[68] B. K. Ayan. "Morphosyntactic Generation of Turkish from Predicate-Argument Structure," *In Proceedings of the COLING Student Session. Association for Computational Linguistics*, 2000, pp. 37-44.

[69] I. Cicekli and T. Korkmaz, "Generation of Simple Turkish Sentences with Systemic-Functional Grammar," *In Proceedings of the Joint Conferences on New Methods in Language Processing and Computational Natural Language Learning. Association for Computational Linguistics*, 1998.

[70] D. Z. Hakkani, "Design and Implementation of a Wide-coverage Tactical Generator for Turkish, a Free Constituent Order Language," Master's thesis, Bilkent University, Turkey, 1996.

[71] M. Temizsoy and I. Cicekli, "A Language-Independent System for Generating Feature Structures from Interlingua Representations,"*In Proceedings of Natural Language Generation. Association for Computational Linguistics*, 1998.

[72] E. Doğan, B. Kaya and A. Müngen, "Generation of Original Text with Text Mining and Deep Learning Methods for Turkish and Other Languages," *In Proceedings of the International Conference on Artiicial Intelligence and Data Processing (IDAP) IEEE*, 2018, pp. 1-9.

[73] A. Güran, N. G. Bayazit and M. Z. Gürbüz, "Eicient feature integration with Wikipedia-based semantic feature extraction for Turkish text summarization," *Turkish Journal of Electrical Engineering and Computer Sciences*, 2013.

[74] M. Kutlu, C. Cığır and I. Cicekli, "Generic Text Summarization for Turkish," *Comput. J. 53*, Oct. 2010.

[75] M. Y. Nuzumlalı and A. Özgür, "Analyzing Stemming Approaches for Turkish Multi-Document Summarization. *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics*, 2014, pp. 702-706.

[76] Ç. C. Birant, Ö. Koşaner and Ö. Aktaş, "A Survey to Text Summarization Methods for Turkish," *International Journal of Computer Applications 144*, 2016.

[77] S. Demir, "Turkish Data-to-Text Generation Using Sequence-to-Sequence Neural Networks," *ACM Transactions on Asian and Low-Resource Language Information Processing*, May. 2022.

[78] T.Verma, Renu, D. Gaur, "Tokenization and Filtering Process in RapidMiner," *International Journal of Applied Information Systems (IJAIS)*, vol. 7, Apr. 2014.

[79] T. Zhan, "Classification Models of Text: A Comparative Study," *2021 IEEE 11th Annual Computing and Communication Workshop and Conference, CCWC 2021*, Jan. 2021, pp. 1221-1225.

[80] Y. Goldberg and O. Levy, "Word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method," *arXiv preprint arXiv:1402.3722*, Feb. 2014.

[81] J. Pennington, R. Socher and C. D. Manning, "GloVe: Global Vectors for Word Representation," *Empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532-1543.

[82] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes and D. Brown, "Text classification algorithms: A survey," *Information* vol. 10, pp. 1532-1543, May. 2019.

[83] T. Aarsen et al., “NLTK,” https://www.nltk.org/, 2020 [Jun. 01, 2022].

[84] X. Zhou, R. Gururajan, Y. Li, R. Venkataraman, X. Tao, G. Bargshady, P. D. Barua and S. Kondalsamy-Chennakesavan, “A survey on text classification and its applications,” *Web Intelligence*, vol. 18, pp. 205-216, 2020.

[85] F. Chollet et al., Keras, https://keras.io, 2015 [Jun. 01, 2022].

[86] T. T. Wong and P. Y. Yeh, “Reliable Accuracy Estimates from k-Fold Cross Validation,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, pp. 1586-1594, Aug. 2020.

[87] M. Grandini, E. Bagli and G. Visani, “Metrics for Multi-Class Classification: an Overview,” *arXiv:2008.05756*, Aug. 2020.

[88] Y. Goldberg, “A Primer on Neural Network Models for Natural Language Processing,” *Journal of Artificial Intelligence Research*, vol. 57, pp. 345-420, Nov. 2016.

[89] A. Jacovi, O. S. Shalom and Y. Goldberg, “Understanding Convolutional Neural Networks for Text Classification, *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Apr. 2020.

[90] D. Mwiti, “NLP Essential Guide: Convolutional Neural Network for Sentence Classification”, https://cnvrg.io/cnn-sentence-classification/, [Jun. 01, 2022].

[91] A. Ozberk and I. Cicekli, “Offensive Language Detection in Turkish Tweets with Bert Models,” *Proceedings - 6th International Conference on Computer Science and Engineering, UBMK 2021*, 2021, pp. 517-521.

[92] T. Rajapakse, “SimpleTransformers”, https://simpletransformers.ai/, May. 2020 [Jun 01, 2022].

[93] Hugging Face, https://huggingface.co/, 2022 [Jun 01, 2022].

[94] J. Alammar, “The Illustrated GPT-2 (Visualizing Transformer Language Models),” https://jalammar.github.io/illustrated-gpt2/, 2022 [Jun 01, 2022].

[95] P. J. Liu, M. Saleh, E. Pot, B. Goodrich, R. Sepassi, L. Kaiser and N. Shazee, “Generating Wikipedia by Summarizing Long Sequences,” *ICLR 2018*, Jan. 2018.

[96] H. Zhang, H. Song, S. Li, M. Zhou and D. Song, “A Survey of Controllable Text Generation using Transformer-based Pre-trained Language Models,” *J. ACM*, Vol. 37, Jan. 2022.

[97] A. Moon, T. Raju and T. Cse, “A survey on document clustering with similarity measures,” *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, pp. 599-601, Nov. 2013.

[98] M. Karakaya, “Sampling in Text Generation,” https://colab.research.google.com, 2021 [Jun. 01, 2022].

[99] A. Fan, M. Lewis and Y. Dauphin, “Hierarchical Neural Story Generation,” *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2018, pp. 889–898.

[100] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever, “Language Models are Unsupervised Multitask Learners,” *OpenAI*, 2019.

[101] M. Karakaya, “turkishReviews-ds,” https://huggingface.co/kmkarakaya, 2022 [Jun. 01, 2022].

[102] C. Caglayan, “PromptGenerator,” https://huggingface.co/cansen88, 2022 [Jun. 01, 2022].

APPENDIX A

Generated Reviews with Miniature GPT

A.1 Uncontrolled Generated Reviews

- otel memnuniyetsizligim ve somestr tatili hakkinda temmuz arasinda izmir optimum avm cika turizm den satin almis oldugumuz fas kraliyet sehirleri tatilimizde giden zamana mi zamana mi yoksa giden zamana mi desem mi yoksa giden zamana mi
- turizm para iadesi yapilmadi ankara dan bir gezi satin aldim tur ucretini odedim fakat iptal etmek icin iade olmadi ucret iadesi yapilmasini istiyorum bir daha gezi planlamistik ve en son gunu toplam borctan dolayi anlasmayla alakalidir kendisi bu sebeple muayeneye
- turizm ile iletisim bilgiler arasinda numarali telefonu paylasmak istiyorum ankara kecioren kardesler motor siparis adresinde yer telefon gecici bir sure hizmet vs ben seyahat etmekte sirketin mentalitesi felaket kotu spor salonunda calisanlarin elleri kurulamak icin kollari bagli sorun nedeniyle rezervasyonu otelin
- ta bugdayin tam kalmasi levent ten bakkaldan eti burcak aldigimda bugdayin biskuvinin tam ortasinda kalmasi hic yakismadi sana eti gerekli aciklamayi bekliyorum muhakkak
- ilac kozmetik sac dokulme sorunu sac ekiminden ay midem bulanmaya devam odalarda sac gibi bir sampuan vardi bir sac gordum ve sac dusunuyorum cunku diger sikayetler sac bile sac bile yoktu bir tane sac dokulme sac ve saclarim saclarim
- eczanesi sancaktepe den sikayetciyim sabah saatlerinde hatay karabaglar nu-

marali fahrettin koca avrupa eczanesine gittim recetemde gozluk yazdi neden var dedigimde bana ilac yazip gonderdi bana ilac yazdi ben sana ilaclari onu bile yazmadi ayrica eczaci karti neden diye sordugumda da da

- otel hizmet memnuniyetsizligi oncelikle ets tur uzerinden gun oncesinde ets ile yaptigim gorusmede otel de oldukca kotu bir oda satin almistim bir otel daha asla yoktu fakat pansiyondan farksiz cok kotuydu en alt seviyede oldugunu ve cok kotuydu en az az

- eczanesi sancaktepe kararina uymama corona sebebi ile ilacimi vermediler kok diye yazilan recete verip gonderdi eczaneye dogru duzgun cevap vermeyip bugun sabah da veremiyoruz deyip beni hasta donduruldu diyor ben ilacini satmaya kalktigimda hicbir degistirmiyor ne giderek iletisim formu formu

- agiz ve dis sagligi merkezi randevu almaya gittigim bir doktor m c ve muayene olmaya gittik doktorun yazdigi halde ve tani kiti yapmadilar bizi baska bir hastanede baska bir otel mi diye soyledi ben baska bir hastanede oldugumu belirtmek istedim istedim

- hotel ayvalik sarimsakli hizmeti kalitesiz hizmet sunuyor otel icin gunluk odeme almislar royal sebaste hotel calisani otel hizmet ama otel yiyecek duzen biri biri ben iki kere temiz bardak canli muzik bile gun ayni seyden gectim en son son

- eczanesi bursa usluplari cok sert virus sebebiyle alinan tedbirler kapsaminda ilac raporlarinin hazirana kadar uzatildigi ve ilacimi eczaneden alabilecegim soylenmisken eczaneye gittigimde kac tane iciyorsun sen kac kez bitirdin seklinde erken bitirdin sen uslubu cok erken odeme plani tane oda oda

- otel turkiye'nin en haftada otel oncelikle otelden memnuniyetsiz ve otel donus dostu oteli tercih ettigimiz oteli hele restoran personel saticinin elinde aldigi parayi hele restoran personel tarafindan hele restoran personel tarafindan rica ettik oncelikle sistemler neden gerekli onlemlerin muayene

- ilac kozmetik sac dokulme problemim nedeniyle onerilerle receteyi kesinlikle birkac kere birkac kere birkac kere tercih ettim birkac kere birkac kere gittim birkac kez arayip hekim bolgesine ay gidip geldim birkac kere form subenize basvurmam gerektigini soylediler birkac hafta sonra sonra

- butik otel de netlesemeyen rezervasyon islemi internet arayisim sonucu internet adana cikisli rezervasyon islemi sonucu aradim kendisinin oranin isletmesi oldugunu soyledi aradim kendisinin oranin yetkilisi oldugunu soyledi samos bolumunun asistani s beye sorduk calisandan gayet icten ve samimi olarak bana bana

A.2 Generated Reviews with Modifying Sequential Input with Topic Technique

- Desired topic : finans

  acil serviste kullanilan malzeme oldugum tl degerinde applecombill adi altinda yuzey ayni zamanda dikmen subesine aldirdim sonuc elde edemedik girisinizi saglayamayiz bende degil mi olur musunuz sormadiniz cikmam gerektigi soylenildi bende mecburen koronavirusten yeni verginin icinde

- Desired topic : finans

  dolayi gelmeyen kart teslimati yapilmamasi mayista ablamla beraber suredir aktif olarak gozuken nhanim bizlere verme gafletinde bulundum fakat ben isten cikarildim odeyecek gucum yeterse bir suru sikayet edecegim olaylari patlak verince kimseye ulasamadim musteri hizmet

- Desired topic : saglik

  saglik ocaginda calisan kiza soru sorarak teshiste bulunup arizanin devam etti ben bir suru para yatirmadan dusunmeniz gerekiyor etrafimdaki vatandaslarin vergisiyle maas alamiyor

- Desired topic : saglik

  alatas elmadag devlet hastanesi nde yedikule gogus hastaliklari hastanesinde ayaktan dakikalik sureden beri hastanedeyim dakika icinde bulundugumuz durum karsisinda saskina donduk receteye istinaden optik ayakkabi aldilar ben denemeseydim oglum bir suru zamanim geldi bir zahmet gonder

- Desired topic : turizm

  dinlenme tesisi elestiriyorum tatilimizi ayinda aldigim kompleksten tek kullanimlik sifre alma sebebim buydu diyebilirim oruclu oldugum halde turnikelerden

ziplayarak uzerime yurudu otel calisanlari lakayit bir suru insan sagligini hice sayiyorlar cocuklar buraya kadar lakayit soyleme

A.3 Generated Reviews with using Modifying Sequential Input with Keywords Technique

- Desired topic : finans

  kredi kayit burosu tarafindan yuklenen puanlari oldugum halde bir miktar para veriyoruz mudure soyledim calismaktan fena sekilde uzlasamiyoruz davamizi gibi koklu bir suru para verip oluyor ve baska bir sey yapamayiz bu sebepten dolayi tanidigimiz adamlar adamlar adamlar bildigimiz tur bildigimiz adamlar bildigimiz basvurusundan adamlar bildigimiz adamlar

- Desired topic : finans

  uzerinden yapilan hayat merkezi sorunuptt kart ucreti sorunu aralik ayi basinda basvurdugum tl kredi kartiyla alisveris yaptim ancak kredi kartim ile donuyorlar bu kadar az once bu sebepten basvurduk bize de tuleren bu sebepten dolayi hicbir sey yapamayiz bu konuda arada durumu durumu arada konuda konuda konuda kadar tarz kadar

- Desired topic : finans

  ziraat bankasi destek kredisi icin muhatap yok nisan tarihinde kapatmama ragmen halen beklemekteyiz bir turlu yaniltmalarla tesise gidip basvurumuzu tamamladik fakat hicbir sekilde hallettik masraflari bir turlu bahaneler uydurarak de

- Desired topic : saglik

  icin muhatap bir zaman kaybi ve travmatoloji bolumunden aldigim lio ms hastaligi yuzunden daha sonra saglik sorunu var deyip duruyor

- Desired topic : turizm

  butik otel ve kotu bulunan caprice deluxe belek antalya kemer melissa residance icerisinde bir suru insan sagligini hafife alinacak kararla tekrar sordum kendisi getirecek bir sey dahil olmak uzere toplam kisi yaralanmistir otelin de

A.4 Generated Reviews with Sampling with Topic Selection Classifier Technique

- Desired topic : finans

  bir uygulama hastanesinde yonetim asla tercih edilen kredi basvurusunda bulunduk o kadar kotu durumda kaldim fakat dun aksamdan ucaga dolayi bir suru insan fotokopisini istiyorum bu surecte degisiklige gidiniz ne yapmam lazim bu durumu anlattigimda onlar yuzunden magdur durumdayiz durumdayim

- Desired topic : saglik

  dolayi goz doktoruna gittim hic kimse yok mart tarihinde saat sularinda antalya erenler hira turizm bakanligi tarafindan getirilen litrelik karton poseti gozume acemice bir sey yapmasaniz bile yoktu ve bunu cikarmadan sustum ancak o tarihlerde basvuran cok fazla bir de sey

- Desired topic : turizm

  bir organizasyon iptali sorunuyaklasik yildir geldigim an birlikte tarihinde cayirbasi sisli mecidiyekoyde market ten sonra seytan ve biz sizi gorebiliyor ve bunu dile getirdigimiz icin bir daha iyi niyetli bir de boyleydi degisen fiyatlarla satiliyorben ve bu durumu cozmek izah

- Desired topic : saglik

  ayinin sonunda ayinda kizimin elinden birakmasi iki gun boyunca odeme yapmak istedigimi defalarca gidip mri dair sms yoluyla bu zamana dogru duzgun verim alamayacaksam son derece rahatsiz oldugumu belirttim bu sebeple de bu sefer de bir sey olmaz yok

- Desired topic : gida

  marketten almis olmama karsin aldigim yere kadar kotu bir supermarketten aldigim yere gittim ve bunu hic acilmamis olmakla basvurdum ama cok sukur ki kagitta not biraktim ve bunun yaninda duran kadin hekimleri birligi de bir daha sonra bu bu

APPENDIX B

Generated Reviews with Multi-layer GPT

B.1 Generated Reviews with Sampling with Topic Selection Classifier Technique

- Desired topic : finans

  1 Nisan 2020'de krediye ait faiz oranı ile kredi kartım onaylandı ancak halen bir sonuç alamadım. Kredi kartımın ödemesi gerçekleştirmiyorlar. 2 aydır değerlendirme aşamasında yazıyor. Bu konuda bu işi böyle oluyor

- Desired topic : finans

  Kredisi. 25/05/2020 de verdiğim faiz ile birlikte bir şekilde kredi kartıma 28 martta faiz oranı ile ile başvuruda bulundum. Fakat hala değerlendirme aşamasında görünüyor ya da olumsuz dönüş yapılmadı. Bunun bu yüzden en

- Desired topic : finans

  22 Nisan tarihinde kredi kartımla başvuru yaptım ama hala sonuç alamadım. Lütfen bir daha ne de olumsuz geri dönüş yok. Artık kredi kartımdan bu kadar saçma sapan bir şey ve zor zor olabilir.

- Desired topic : saglik

  Sorunu. Bursa Büyükşehir Belediyesi Covid19 virüsü yüzünden bir aylık evde yaşıyoruz. 6 aylık en yüksek fatura nasıl bir tutar bu faturayı bu kadar yüksek bir düzen ediyorsunuz. Siz nasıl bir saçmalık olmaz bir şey olamaz çok

- Desired topic : saglik

Etmesi ve Randevu. 15 gün önce randevu aldım. İptal etmek için 1 hafta sonra aradım. Çağrı merkezi cevap vermiyor. En son bir saat oldu. Telefon da açan yok. Çağrı merkezi telefonu cevap vermiyorlar.

- Desired topic : saglik

  Eksikliği. İstanbul Medipol Hastanesi Doktor Doktor Sorunu. Dr. Dr Ö*** bey, doktorun hastanede hasta etti ve tahlil yok. Bugün 2 saat sıra bekletip doktor yok, doktor. Bu yüzden 2 hafta önce sabah

- Desired topic : turizm

  Otel otelde kaldım sonra bize iki gece bir gece 3 kere servis geldi biz 1 gün sonra randevu veren yer bir tur bile değil o günden bu yana. Ayrıca gece bir gün sonra aradım.

- Desired topic : turizm

  ETS Tur ile beraber İstanbul turu almış olduğumuz tatil ile ilgili kişi için bilgi verilmeden iptal etti. Ben de bu zamana kadar kötü0bir bir şekilde paramı iade bekliyorum bu durum ki.

- Desired topic : turizm

  ETS Tur sitesinden aldığım Karadeniz turuna iptal ettirmek istedim. Otelden 5 gün sonra, 4 gün bekledim. Bir gün içerisinde dönüş yapmadılar. Çağrı merkezine ulaşmak mümkün değil. 2 kişi daha önce arayıp.

- Desired topic : spor

  spor salonu tam bir kötü yani bu kadar kötü bir yemek olamaz. D-Smart daha önce aldık ama hiç memnun kalmadım. Hiç böyle bir sorun olmamasına rağmen hiçbir dönüş olmadı. Bir daha böyle bir marka görmedim

- Desired topic : spor

  bir spor salonu var. 2 kere de üye olduk fakat hiç bir türlü bilgi alamamaktayım. Bu durum hakkında müşteri hizmetlerine yazıyorum açan sıfır. İlk başta bu Yana bir kez mail attım cevap veren yok

- Desired topic : spor

.com Üyelik İptal, Paramı İade Etmiyor. Ben bu süreçte üye oldum. Bir ay önce bir üyelik yaptırdım. Bana iptal ettiler. Parayı iade olmadı. Hiçbir dönüş yok. İnternet siteleri üzerinden da üye oldum.

- Desired topic : gida

  Fiyat Fiyatları. Marketten aldığımız 15 lira aldığım 2 TL fiyatı var. Ben. Bu kadar fazla bir fiyat görmedim! Her gün aynı yerde 50 TL civarı bir zam geldi. Çok kısa bir süre sonra bu kadar

- Desired topic : gida

  Geç Teslimat Sorunu. Ankara'da bulunan 2.5 litrelik sipariş etmiş olduğum siparişin kargoya verilmesini istiyorum. 5-10 iş günü içerisinde teslim edileceği halde. Bugün 13 Şubat'ta teslim edilmiş. Bu nedenle

- Desired topic : gida

  Gelen Gibi Firma Olması. Merhaba, gelen giden beyaz deterjan suyu yıkamalerinde kokusu olarak geliyor, iki tane halı kullandım ve bu şekilde da koku yapıyor. Ayrıca hiç memnun kalmadım, bu kadar kalitesiz, büyük bir