INTERVIEWSTER: A CHATBOT EVALUATING COMPETENCY BASED INTERVIEWS USING TRANSFORMER MODELS

ONUR EMRE ATICI

MEF UNIVERSITY

JULY 2022

MEF UNIVERSITY
GRADUATE SCHOOL OF SCIENCE AND ENGINEERING
MASTER'S IN INFORMATION TECHNOLOGIES

M.Sc. THESIS

INTERVIEWSTER: A CHATBOT EVALUATING COMPETENCY BASED INTERVIEWS USING TRANSFORMER MODELS

Onur Emre ATICI
ORCID No: 0000-0003-0939-5582

Thesis Advisor: Assist. Prof. Dr. Şeniz DEMİR

JULY 2022

ACADEMIC HONESTY PLEDGE

I declare that all the information in this study, is collected and presented in accordance with academic rules and ethical principles, and that all information and documents that are not original in the study are referenced in accordance with the citation standards, within the framework required by the rules and principles.

Name and Surname: Onur Emre ATICI

Signature:

ABSTRACT

INTERVIEWSTER: A CHATBOT EVALUATING COMPETENCY BASED INTERVIEWS USING TRANSFORMER MODELS

Onur Emre ATICI

M.Sc. in Information Technologies

Thesis Advisor: Assist. Prof. Şeniz DEMİR

July 2022, 50 Pages

Recruitment is one of the most verbal and communicative area of Human Resources (HR). This department has many aspects that is open to innovation but also open to the bias, because of the dominance of the human factor. This brings the need (and chance) of many innovation possibilities coming up with the progress in artificial intelligence technologies. Here we will focus on creating a chatbot named as "Interviewster" that welcomes candidates, gathers information (such as name-surname, work status, computer knowledge, education, hobbies), and provides competency-based questions about their past experience and helps them to answer these questions correctly. This chatbot welcomes a candidate and starts the conversation, saves the data collected from the candidate, conducts a competency based interview and decides if candidate has the required competency or not by using natural language processing techniques that utilize neural network architectures and transformer-based technologies. The chatbot is running on the web, coded in python, published to the web by Flask, works on a python core with a Mysql database.

In this thesis, interview practices are first introduced, and the methods and applications of competency-based interviews are described. The architecture of the chatbot named as Interviewster is then explained by providing details of the technologies, libraries and machine learning techniques being used. Finally, the results of an evaluation study where the transformer-based models BERT, DeBERTa, and ELECTRA models are applied to the competency-based interview results of real candidates are discussed in detail.

Keywords: NLP, BERT, DeBERTA, ELECTRA, Interviews, Machine Learning, Deep Learning, Classification

Numeric Code of the Field: 92416

ÖZET

INTERVIEWSTER: TRANSFORMER MODELLERİNİ KULLANARAK YETKİNLİK BAZLI MÜLAKAT YAPAN BİR SOHBET ROBOTU

Onur Emre ATICI

Bilişim Teknolojileri Yüksek Lisans Programı

Tez Danışmanı: Dr. Öğr. Üyesi Şeniz DEMİR

Temmuz 2022, 50 Sayfa

İşe alım, insan kaynaklarının en sözel ve iletişimsel alanlarından biridir. Bu departmanın insan faktörünün baskın olması nedeniyle yeniliğe açık olduğu kadar önyargıya da açık olan birçok yönü bulunmaktadır. Bu da yapay zeka teknolojilerindeki ilerlemeyle birlikte birçok inovasyon ihtiyacını (ve şansını) beraberinde getirmektedir. Bu çalışmada adayları karşılayan, bilgi toplayan (ad-soyad, iş durumu, bilgisayar bilgisi, eğitimi, hobileri gibi) ve geçmiş deneyimleri hakkında yetkinlik bazlı sorular sunan ve bu soruları doğru cevaplayabilmesi için onlara yardımcı olan "Interviewster" adlı bir sohbet robotru oluşturmaya odaklanılmaktadır. Bu sohbet robotu adayı karşılar ve konuşmayı başlatır, adaydan toplanan verileri kaydeder, yetkinlik bazlı görüşme yapar ve sinir ağları mimarileri ve transformer tabanlı teknolojileri kullanan doğal dil işleme teknikleri ile adayın gerekli yetkinliğe sahip olup olmadığına karar verir. Web üzerinde çalışmakta olan bu sohbet robotu Python ile kodlanmış ve Flask ile web'de yayınlanmış olup Mysql veritabanını kullanan bir Python çekirdeği üzerinde çalışmaktadır.

Bu tezde ilk olarak mülakat uygulamaları tanıtılmakta ve yetkinlik bazlı mülakatların yöntem ve uygulamaları anlatılmaktadır. Sonrasında Interviewster olarak adlandırılan sohbet robotunun mimarisi, kullanılan teknolojiler, kütüphaneler ve makine öğrenmesi teknikleri, detayları verilerek açıklanmıştır. Son olarak da transformer tabanlı modeller olan BERT, DeBERTa ve ELECTRA modellerinin gerçek adayların yetkinlik bazlı mülakat sonuçlarına uygulandığı bir değerlendirme çalışmasının sonuçları detaylı olarak tartışılmıştır.

Anahtar Sözcükler: NLP, BERT, DeBERTA, ELECTRA, Mülakatlar, Makine Öğrenmesi, Derin Öğrenme, Sınıflandırma

Bilim Dalı Sayısal Kodu: 92416

FOREWORD

I would like to thank my thesis advisor Assist. Prof. Dr. Şeniz Demir, for her intense support during the thesis process and for sharing her valuable time and knowledge with me. I would also like to thank my jury members Assist. Prof. Dr. Tuna Çakar and Assist. Prof. Dr. Didem ABİDİN contributed to my education. I am very thankful to my family for always being by my side and supporting me with their endless trust and love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AI	: Artificial Intelligence
ANN	: Artificial Neural Networks
BERT	: Bidirectional Encoder Representations from Transformers
CBI	: Competency Based Interview
DEBERTA	: Decoding-Enhanced BERT with Disentangled Attention
ELECTRA	: Efficiently Learning an Encoder that Classifies token Replacements Accurately
HR	: Human Resources
IDC	: International Data Corporation
ML	: Machine Learning
NLP	: Natural Language Processing
PML	: Pre-trained Machine Learning
RNN	: Recurrent Neural Network
SQL	: Structured Query Language
S.T.A.R.	: Situation - Task - Action - Result
TF_IDF	: Term Frequency-Inverse Document Frequency

INTRODUCTION

Chatbots have become an important part of our lives with the rise of artificial intelligence (A.I.). Chatbots, which we encounter in many areas from shopping sites to banking, found an important place especially in our mobile phones and entered our lives as our personal assistants. Unfortunately, human resources, which is the most humane area of business life, has not yet been integrated into the chatbot world enough. Especially the companies that carry out their recruitment processes in English have made significant gains in the candidate evaluation process by using psychometric tools that work with the support of advanced natural language processing algorithms. Currently, there are companies in the market that implement assessment center applications through a chatbot-like structure. However, since interviews and especially competency-based interviews are assessment tools that should include a certain interaction, there are not many solutions in this area. In order to close this gap, we tried to develop a chatbot called Interviewster as an academic effort.

An interviewer chatbot had to have a certain interaction with the candidates, as well as be able to conduct the interviews and assess them as accurately as possible. In order to do this, we have established a web-based chatbot infrastructure. We tried to increase the dialogue quality by supporting this infrastructure with tensorflow. to evaluate competency based question answers, we used PML structures based on transformer technology like BERT, which we think is a very important gain in natural language processing.

The aim of this study will be to contribute to the development of recruitment technologies and to provide a reference point for academicians or entrepreneurs who want to produce solutions in this field in the future.

1. RECRUITMENT INTERVIEWS

1.1. Interviews in Recruitment Process

Interviews are the key element of a hiring process where the decision of “yes” or “no” is given to a candidate. Interview is basically a face to face conversation with the job candidate in order to assess mainly soft skills and competencies. Even nowadays the interview process evolved to a digital way, the main idea of meeting in person is still the best practice. However, to fill a standard empty position, a recruiter or a recruiting manager has to meet 5 to 10 candidates, with no guarantee of success because of untalented recruiters, the lack of skillful candidates, or bias and halo effect.

1.2. Interview Types

1.2.1. Standard Interviews

Interviews are the key stages of a hiring action. In a standard interview there is generally one hiring manager and one candidate. There are also different approaches like meeting with more than one candidate at once. There is a version called panel interviews where many hiring managers meet a candidate at the same interview. Even this is a very bad practice for the candidate because it will be impossible to concentrate to the interviewer, it’s a great way to avoid never ending interviews with every manager related to the job.

For a successful interview experience, both the candidate and the recruiter must be calm and comfortable. The recruiter has to be comfortable because he/she has to concentrate on candidates' words, mimics, and the body language. Candidate has to be comfortable because introducing himself/herself to someone else is a very stressing process especially if the goal is to get accepted to a job or a master’s program. After Covid-19 pandemics created a revolution in remote working, interviews, at least at the first phase of hiring process, is started to being conducted online with different online meeting tools. This gives candidates more comfort but takes recruiters advantage to observe candidates face to face.

1.2.2. Competency Based Interviews

Competency Based Interviews (CBI) are well structured interviews to assess the candidate's competencies which are needed for an open position. In this type of interviews, the interviewer has a set of questions for each skill/competency which will be assessed. In daily practice, interviewer can ask the same question with different wordings to make the candidate understand the question clearly.

CBI depends on the principle that the past behavior of a candidate is an indicator for the performance that he/she will show in the future. The interviewer asks a question about a skill and tries to understand how the candidate might behave in a situation that he/she will experience in his/her job.

The well-known method for a successful interview is the STAR method. Below we describe this method briefly:

- S – Situation: The situation that the candidate has to deal with
- T – Task: The task given to the candidate (if any)
- A – Action: The action that the candidate takes
- R – Result: The result of the action, what the candidate learns from the event[1]

While using the STAR method, the interviewer will help the candidate to express a behavior related to the competency being assessed in a logical way. Here is a few examples of CBI questions and sub questions:

> *"Tell me about a situation that you experienced in your past work or in your daily life, that you had to handle a very complicated job which you may not know how to do or never experienced the same situation before?*
>
> *Sub Questions:*
> *- What was the situation?*
> *- How did you managed to finish the job?*
> *- What difficulties did you face?*
> *- How was the feedback from your managers when you finish the job?"*

[1]Retrieved from nationalcareers.service.gov.uk: https://nationalcareers.service.gov.uk/careers-advice/interview-advice/the-star-method (2022)

"Give an example of a time when you had to make a difficult decision in jour job or in your daily life before

Sub Questions:
- What was the decision about?
- Why did you make that decision?
- What was the consequences?"

"Tell me about a situation in which you were working as part of a team. How did you make a contribution?

Sub Questions:
- What was the situation?
- How did you make a contribution?
- How was the feedback from your teammates?

The interviewer tries to get answers for all sub questions. If he/she couldn't, the interviewer will ask sub questions in order to collect a clear evidence about candidate's past behavior.

1.3. Competencies

The concept of competency was first used by Boyatzis (Boyatzis 1982). Boyatzis has developed success factors for managers in different areas. He defined these factors in terms of individual quality, self-confidence, experience, different style behaviors, and characteristics. Wooruffe (1993) defined competency as a behavioral factor that affects the job performance.

Competencies may vary by company, department, and role. Although the name is the same, the definitions of these competencies may vary. Below you can find a few examples of the most used competencies:

- Success-Oriented
- Result Oriented
- Analysis Power
- Strategic Thinking
- Creativity
- Analysis Capability
- Team Management
- Leadership
- Building Relationships
- Communication
- Adapting to Change
- Coping with Stress

- Planning
- Control

Below you can find some competencies that Koç Holding determines for its companies (Öztürk, 2010):

- **Teamwork:** Leads to a common goal, contributes to the team success with solidarity, sustains high motivation in the team and provides commitment.
- **Communication**: Listens carefully, expresses himself/herself effectively, is successful in negotiating, resolving disagreements and persuading other parties, develops long-term relationships.
- **Result Orientation:** Makes effective decisions quickly by taking calculated risks and using initiative, and achieves results by showing determination in the case of experienced barriers and uncertainties.
- **Creativity and Entrepreneurship:** Sees opportunities by following innovations, thinks outside the box, creates a difference, analyzes and takes action by rapidly evaluating creative ideas.

1.4. Measuring Success in CBI Interviews

In competency-based interviews, the success is measured using a 5-point or 13-point scale. For example, if we consider a 5-point scale, 1 point is given to a candidate if he/she can demonstrate the competency at the lowest level, and 5 points is given if the candidate demonstrates the highest level of competency. The detailed scale is given below:

- **1 point:** The candidate failed to provide any example of competency, or the examples that he/she gave were of very low quality or not relevant to the competency.
- **2 points:** The candidate was able to give a limited number of examples about the competency. The evidences he/she gave are of poor quality. He/she will rarely exhibit this competency in business life.
- **3 points:** The candidate was able to give sufficient examples related with the competency. The quality of the samples he/she gave is at an acceptable level. He/she will show this competency situationally in business life.

- **4 points:** The candidate was able to give many examples about this competency. The quality of the samples he/she gave is high. He/she will be able to demonstrate this competency mostly in business life.
- **5 points:** The candidate was able to give more examples of the competency than expected. The quality of the samples he/she gave is high. He/she will always show this competency in business life.

While companies are evaluating candidates based on these competencies, they can set a threshold or evaluate all competencies together in order to identify successful candidates. The important thing in evaluation is that the evaluator has the necessary training and experience in this regard. If not, competency-based interview evaluations may produce incorrect results.

1.5. The Bias Problem In Candidate Assessment Process

According to a research of Ozen & Kizildag (2018) it was observed that interview errors are caused by common mistakes:

- Focusing on negative information in the candidate's resume before the interview,
- Prejudice during the interview due to the first impression or Halo effect
- Getting affected by the candidate's body language more than it should
- Comparing the candidate with himself/herself
- Comparing the candidates with each other
- Asking guiding questions

These bias related problems are not strictly related with the assessor's experience. Even the most experienced assessors can make these mistakes because of many different reasons like job related, motivational or personal problems. Even a bad memory or a similarity with someone that the recruiter does not like can cause this.

2. CHATBOTS AND THEIR USAGE IN HUMAN RESOURCES

2.1. Chatbots in a Digital World

IBM defines chatbot as "*a computer program that uses artificial intelligence and natural language processing to understand customer questions and automate responses to them by simulating human conversation*" (Chatbots Explained, 2020). These virtual agents, often known as chatbots or AI assistants, are becoming more and more common across many industries. But as chatbots go by many various names, they change according to their levels of intelligence. The International Data Corporation (IDC) projects that by 2022, focusing on cognitive and AI systems would have increased by more than three times from the $24.0 billion predicted for 2018 (Chatbot Trends Report 2021, 2021).

Businesses may utilize AI-based chatbots to comprehend customer behavior, purchase patterns, and preferences over time and respond to enquiries accordingly. One of the key aspects driving the market growth is chatbots' benefits over traditional forms of customer care. Chatbots may be included into a variety of user interaction channels, including websites, email, SMS, and messaging programs. They gather client information from databases and customer service exchanges in order to give them a tailored experience. Additionally, they are capable of identifying human emotions including wrath, perplexity, fear, and joy (Chatbot Trends Report 2021, 2021).

2.2. Artificial Intelligence in Chatbots

AI chatbots are chatbots trained to have human-like conversations using natural language processing. With NLP, an AI chatbot is able to interpret human language as it is written, which enables them to operate more or less on their own. In other words, an AI chatbot software can understand language outside of pre-programmed commands and provide a response based on existing data. This allows site visitors to lead the conversation and voice their intent in their own words (drift.com, 2022). A simple working schema of an AI chatbot is given in Figure 2.1.

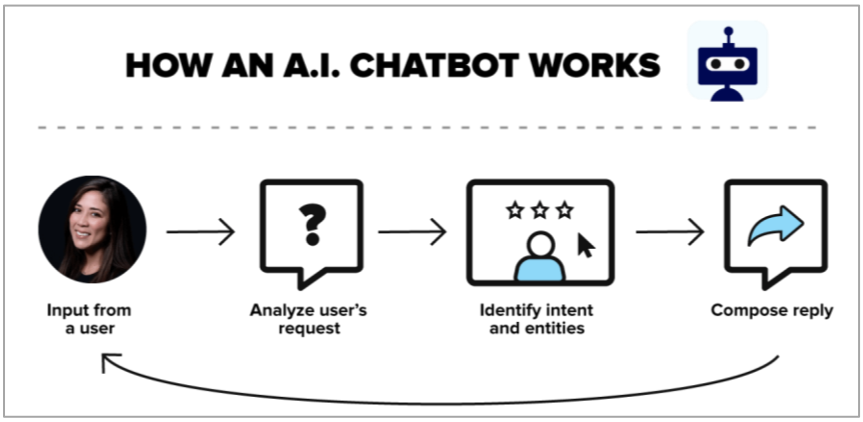


Figure 2.1 How an A.I. chatbot works. A.I. chatbots takes inputs from the user, analyze the request with Machine Learning algorithms and compose a reply. The main idea is to mimic a human responses and make user to think he/she is interacting with a real person (drift.com, 2022)

Using predetermined chat scripts, a database of responses, and advanced machine learning, a chatbot may identify recurrent patterns in talks with people. The chatbot also "learns" from a database, which makes it more advanced. A chatbot that uses machine learning, for instance, may provide updates and personalized alerts, respond in real time to customer inquiries, and assist users in finding items and services on a website.

These deep learning chatbots can mimic human speech and require less human interaction. Deep learning chatbots use structured data and human-to-human conversation to make judgments by building numerous layers of artificial neural networks. An essential element of the AI chatbot algorithm is the NLP layer, which allows computer programs to translate and mimic human conversation through predictive analytics and sentiment analysis along with text classifications.

A chatbot, which determines the precise words and actions, must react to a user's input by dissecting a query into entities and intents. For instance, inquiries like "I wish to place an order for a bag. Do you offer bags? I'd want to get one" should be interpreted correctly by a chatbot algorithm so that the user is allowed to view the bag alternatives available on a website (Iuchanka, 2022).

In human resources technologies ecosystem, there are different types of chatbots which help candidates to get information about the company, help new employees to onboard, conduct case based assessment and psychometric tools for hiring. There are some interview chatbots in the literature but they were all designed for the English language. Our chatbot is the first of its kind since it can conducting a competency based interview in Turkish.

2.3. Human Resources Chatbots

Jaro addressed common concerns that a candidate faces when it comes to attend mass interviews (Purohit, J., Bagwe, A., Mehta, R., Mangaonkar, O., & George, E. ,2019). The list of difficulties includes inconsistent interview questions, various days and hours of the day, the interviewer's attitude, the location of the interview, and so on. By suggesting a chatbot that conducts interviews by evaluating the CV of candidates, JARO expedited the interview process towards an objective decision-making process. The chatbot then develops a series of questions to be asked to the candidate. The technology has functions like automated interviewing and resume analysis. Using an NLP model, which is useful in this process, the machine would additionally ask questions depending on the candidate's prior replies. The program would examine the data gathered during the interview process to find the best candidate for the position being offered.

In the study of Suakanto et al. (Suakanto, S., Siswanto, J., Febrianti Kusumasari, T., Reza Prasetyo, I., & Hardiyanti, M. (2021)), artificial intelligence (AI) or machine learning is used to develop a chatbot that can conduct interviews and interpret the results. The key point was that human-driven interviewing is still not scalable to big numbers and can introduce bias. The system has saved the results in order for a machine learning system or a human expert to evaluate or understand them.

Grabjobs.com's interview chatbot screens the applicants as they apply, scores and ranks them according to their qualifications to the role, identifies which applicants the recruiter should assess first (grabjobs.co, 2022). Every applicant goes through a few question to complete their application which are designed to gather the basic information such as experience, education, skills, geographic location, and availability.

Siabot's interview chatbot assistant can make technical pre-screening, automatic scheduling, asking response, skill and experience based questions based questions, face/gesture recognition and AI based technical scoring (siabot.com, 2022).

3. INTERVIEWSTER'S CHATBOT FRAMEWORK

3.1. Architecture

Our chatbot, Interviewster, consists of two different component. The first component interacts with the user, establishes a dialogue and collects the responses. The second component evaluates these responses and checks whether the person has the relevant competency.

The interaction component runs on a web interface that works with Python and Flask[2]. Flask is a micro web framework that helps creating web applications in Python. The component, which is specially prepared for the interviewer - candidate dialogue and trained with Tensorflow[3], performs the process of meeting the candidate, introducing the process, starting the interview and collecting the answers with a chatbot interface. The decision component evaluates the candidate's answers in the background.

The chatbot technology of Interviewster uses SQL and Python libraries. We used the NLTK[4] library to clean stopwords and tokenize the sentences. We also used a Python based lemmatizer called "Zeyrek"[5] (Bulat, 2022). Besides we used a Python based morphological analyzer of Turkish language which is provided by the Zemberek-NLP toolkit (Akın, 2022)[6]. The chatbot uses a MySQL database to store interview questions and candidate answers.

The architecture of the Interviewster is given in Figure 3.1.

---

[2] https://flask.palletsprojects.com/
[3] https://www.tensorflow.org/
[4] https://www.nltk.org/
[5] https://github.com/obulat/zeyrek
[6] https://github.com/ahmetaa/zemberek-nlp

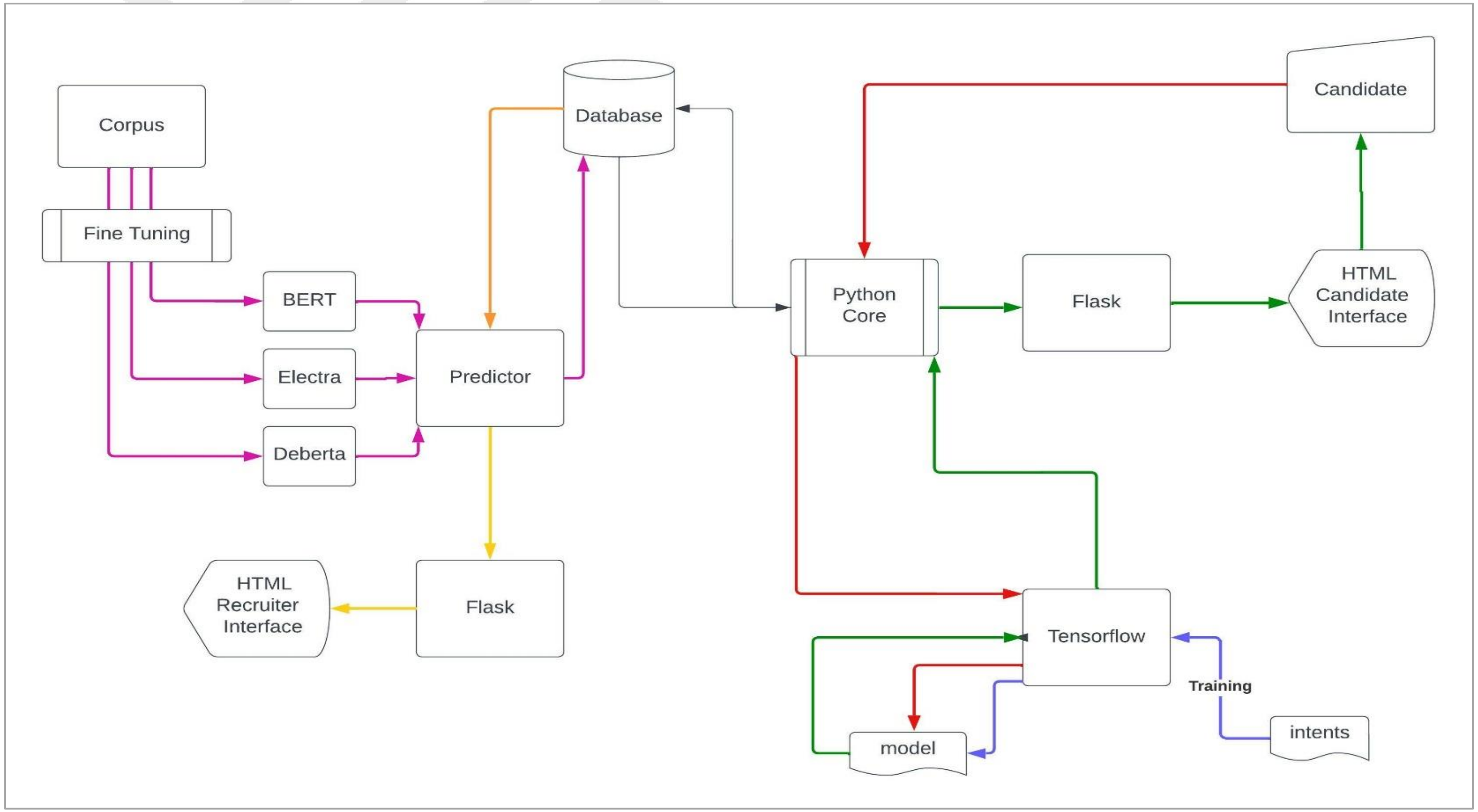


Figure 3.1 Interviewsters Architecture. The Architecture of Interviewster consists of two different components. The first component which interacts with the user, based on Python and Flask for web framework and tensorflow for machine learning. The second component which evaluates candidate responses uses PML models.

In Figure 3.2 some screenshots of Interviewster candidate interface while conducting an interview is given. At the start of the conversation, the chatbot talks with the candidate for a while as an ice-breaker. After the candidate wants to start the interview, the chatbot starts asking competency based interview questions one-by-one (Some parts of the questions are hidden because they are similar to the real questions.)

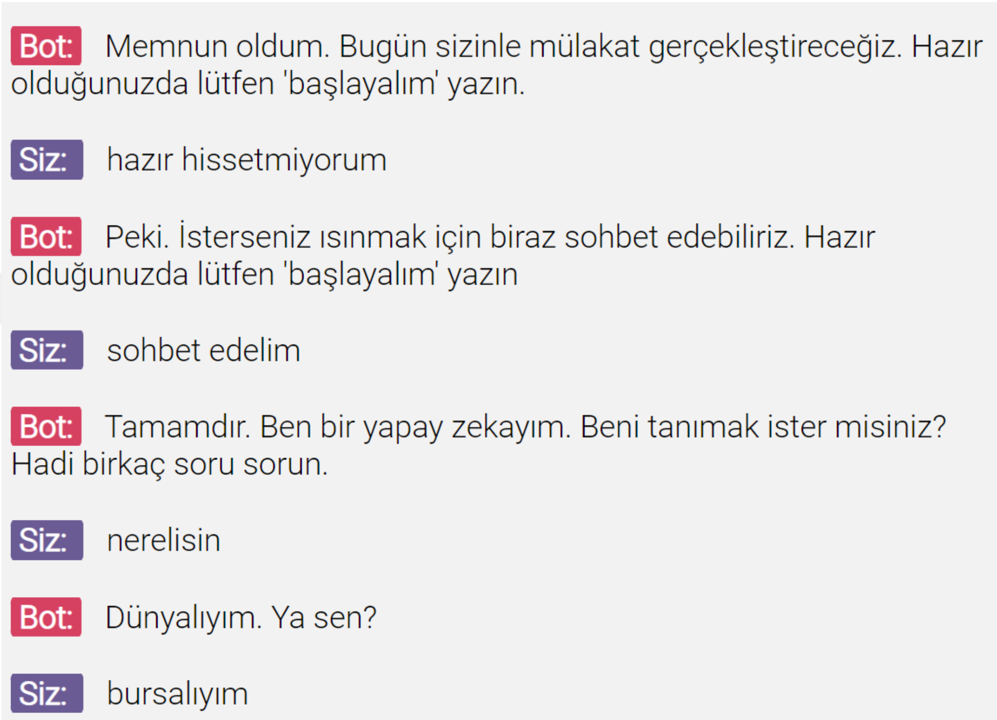
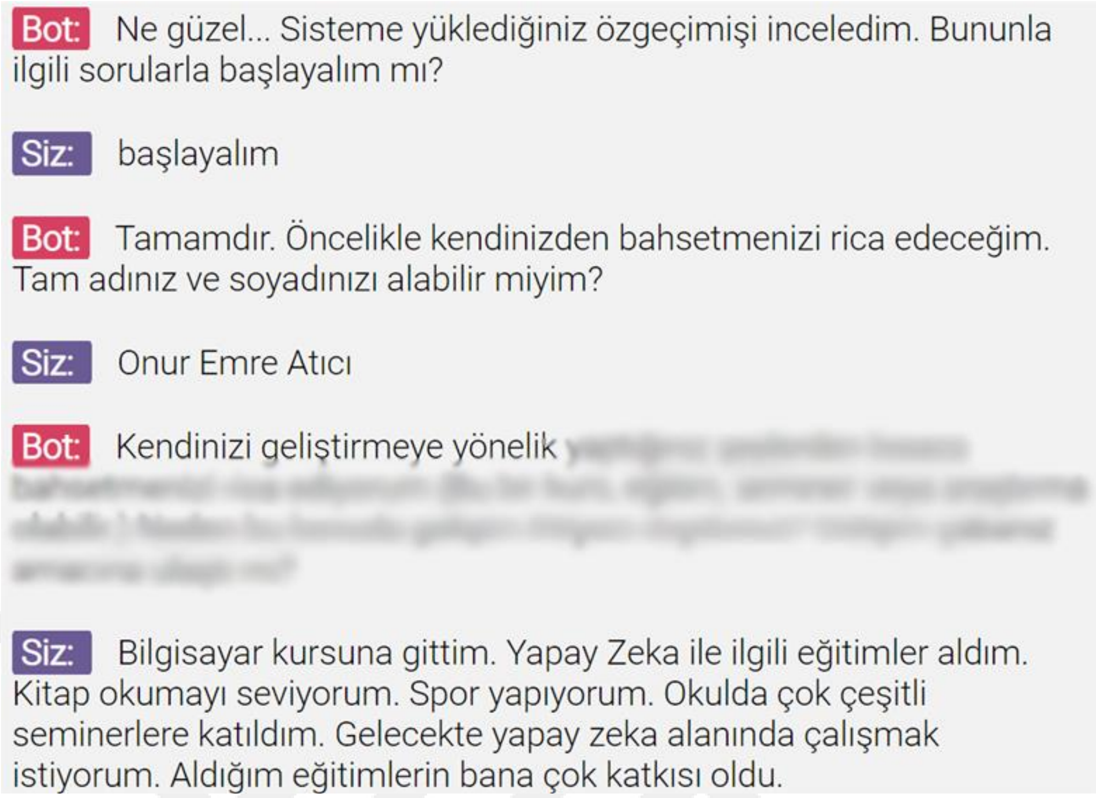


Figure 3.2 Screenshots from a full interview with a candidate



Bot: Ne güzel... Sisteme yüklediğiniz özgeçimişi inceledim. Bununla ilgili sorularla başlayalım mı?

Siz: başlayalım

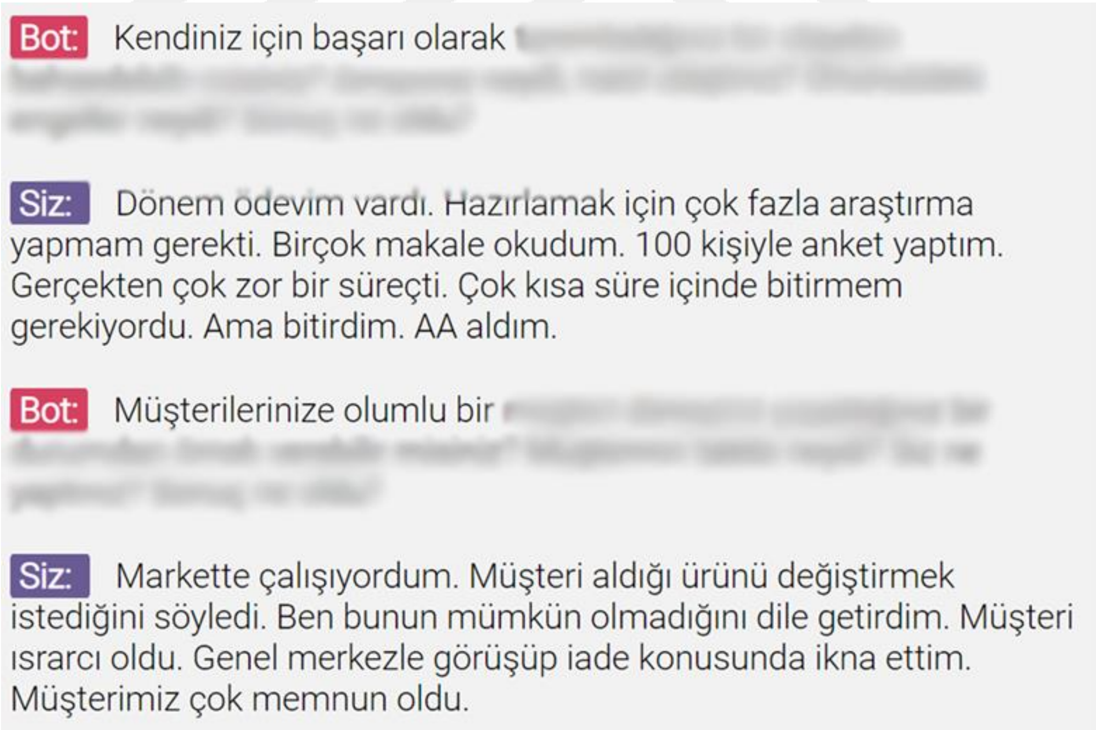
Bot: Tamamdır. Öncelikle kendinizden bahsetmenizi rica edeceğim. Tam adınız ve soyadınızı alabilir miyim?

Siz: Onur Emre Atıcı

Bot: Kendinizi geliştirmeye yönelik

Siz: Bilgisayar kursuna gittim. Yapay Zeka ile ilgili eğitimler aldım. Kitap okumayı seviyorum. Spor yapıyorum. Okulda çok çeşitli seminerlere katıldım. Gelecekte yapay zeka alanında çalışmak istiyorum. Aldığım eğitimlerin bana çok katkısı oldu.

Figure 3.2 Continued (1)



Bot: Kendiniz için başarı olarak

Siz: Dönem ödevim vardı. Hazırlamak için çok fazla araştırma yapmam gerekti. Birçok makale okudum. 100 kişiyle anket yaptım. Gerçekten çok zor bir süreçti. Çok kısa süre içinde bitirmem gerekiyordu. Ama bitirdim. AA aldım.

Bot: Müşterilerinize olumlu bir

Siz: Markette çalışıyordum. Müşteri aldığı ürünü değiştirmek istediğini söyledi. Ben bunun mümkün olmadığını dile getirdim. Müşteri ısrarcı oldu. Genel merkezle görüşüp iade konusunda ikna ettim. Müşterimiz çok memnun oldu.

Figure 3.2 Continued (2)

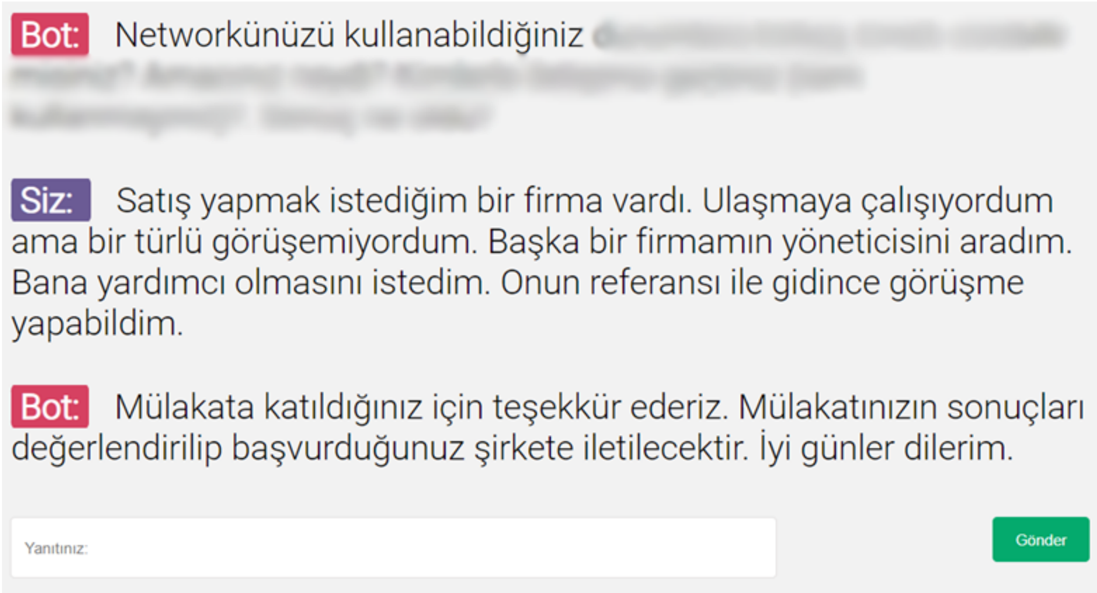


Figure 3.2 Continued (3)

3.2. The Interaction Component

The interaction component mainly uses an end-to-end open-sourced machine learning framework based on tensorflow (tensorflow.org, 2022). It is distributed as a numerical computation framework and released by Google. Its main focus is to alleviate the details of implementing a neural network. Currently tensorflow has around 3,000 contributors and more than 125,000 commits (Ganegedara, 2018).

This component uses a sequential tensorflow keras model to understand users' requests, start communication and conduct interviews. In order to understand users' requests, we used an intent file that we created with main dialogs taken from real candidate-interviewer communications before or during an interview. The interviews were made by nearly **50** different junior and senior, but experienced interviewers so the dataset can be considered as almost unbiased. All candidates were between 21 – 30 years old and the job position was about sales. A total of **1901** interview comments were included in the model. Although there was much more data, this is the amount of data we have left after cleaning the data, removing the comments which are near the limit score of positive and negative, and balancing data on the basis of competency. A proper candidate answer includes the "star" method elements (i.e., situation, task, action and result) which are described in Section 1.2.2.

3.3. Tensorflow and Artificial Neural Networks

A group of neurons that are physically joined together make up a biological neural network. The overall number of neurons and connections between them can be quite high, as can the number of neurons that each neuron is linked to. Cognitive modeling and AI make an effort to replicate brain networks. It is made up of many neurons that link the input set and output (Parvizi & Khishe, 2020).

Information processing patterns known as Artificial Neural Networks (ANNs) are created by replicating biological neural networks, such as the human brain. They are created by effective internal communications that cooperate to address particular issues. By processing experimental data, ANNs "learn" the knowledge or rule underlying the data and transfer it to the network structure. The capacity to learn is the most crucial trait of an intelligent system. A system that can learn is more adaptable and simpler to program, making it better suited to handle new equations and problems. As human brain's creativity, adaptability, and parallel processing specialties are fascinating, machines would benefit greatly from having these qualities as well. An ANN is a unique data processing system that derives from the human brain and processes data into small and very large processors. There are numerous deposits that operate in parallel and networked fashion to address issues. With the use of programming skills, a data structure that can function as a neuron is created for these networks. The node structure refers to this. Then, they build a network between these nodes and use a training method to train the network. Each connection (synapses or connections between nodes) in this neural network or memory has a weight, and the nodes can be in either of two active states (on or off) or inactive states (off or 0). The next inactive node is stimulated or activated by positive weighted connections, and the following inactive node is inhibited or inactivated by negative weighted connections (if active).

Tensorflow is an end-to-end open source platform for machine learning and a Python-friendly open source library for developing neural networks It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. Tensorflow operates on multidimensional arrays or tensors represented as tf.Tensor objects (tensorflow.org, 2022). In tensorflow tensors are units of data and ANN’s are functions which can process the data stored in tensors.

We used a tensorflow model in order to classify the intents of a user during the interaction. We trained our model with 30 intents which are necessary for an icebreaker conversation with a candidate. Figure 3.3. presents some of the intent classes that we utilized in our system. Every intent class has a unique label defined as "tag". The "patterns" are possible utterances (sentences) that a candidate might use during the conversation. These utterances are associated with the corresponding intent class. The "responses" are possible answers that the chatbot will give if the candidate’s input is classified with the corresponding intent class. The "context_set" is used to define a function that will be triggered if the corresponding intent is identified. We'll leave adding more intent classes to our system as future work since more than 250 intents are needed to create a good model for commercial purposes.

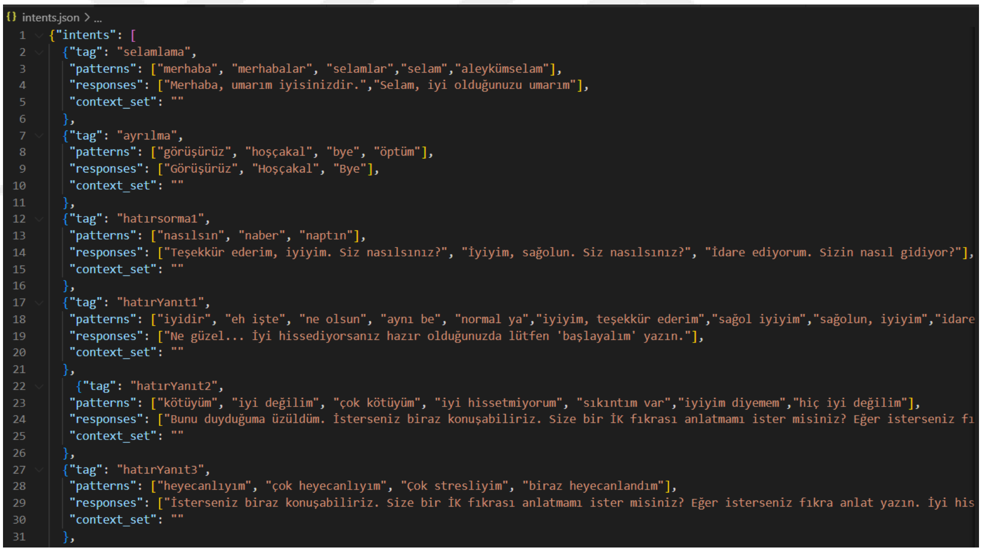

intents.json > ...
```
{"intents": [
  {"tag": "selamlama",
   "patterns": ["merhaba", "merhabalar", "selamlar","selam","aleykümselam"],
   "responses": ["Merhaba, umarım iyisinizdir.","Selam, iyi olduğunuzu umarım"],
   "context_set": ""
  },
  {"tag": "ayrılma",
   "patterns": ["görüşürüz", "hoşçakal", "bye", "öptüm"],
   "responses": ["Görüşürüz", "Hoşçakal", "Bye"],
   "context_set": ""
  },
  {"tag": "hatırsorma1",
   "patterns": ["nasılsın", "naber", "naptın"],
   "responses": ["Teşekkür ederim, iyiyim. Siz nasılsınız?", "İyiyim, sağolun. Siz nasılsınız?", "İdare ediyorum. Sizin nasıl gidiyor?"],
   "context_set": ""
  },
  {"tag": "hatırYanıt1",
   "patterns": ["iyidir", "eh işte", "ne olsun", "aynı be", "normal ya","iyiyim, teşekkür ederim","sağol iyiyim","sağolun, iyiyim","idare
   "responses": ["Ne güzel... İyi hissediyorsanız hazır olduğunuzda lütfen 'başlayalım' yazın."],
   "context_set": ""
  },
   {"tag": "hatırYanıt2",
   "patterns": ["kötüyüm", "iyi değilim", "çok kötüyüm", "iyi hissetmiyorum", "sıkıntım var","iyiyim diyemem","hiç iyi değilim"],
   "responses": ["Bunu duyduğuma üzüldüm. İsterseniz biraz konuşabiliriz. Size bir İK fıkrası anlatmamı ister misiniz? Eğer isterseniz fı
   "context_set": ""
  },
  {"tag": "hatırYanıt3",
   "patterns": ["heyecanlıyım", "çok heyecanlıyım", "Çok stresliyim", "biraz heyecanlandım"],
   "responses": ["İsterseniz biraz konuşabiliriz. Size bir İK fıkrası anlatmamı ister misiniz? Eğer isterseniz fıkra anlat yazın. İyi his
   "context_set": ""
  },
```

Figure 3.3 Some of the intents used in our model

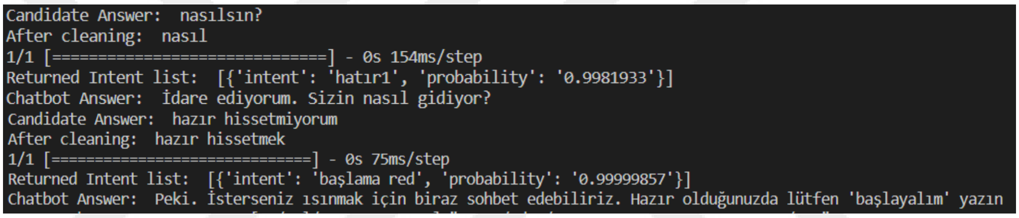
The details of the intent classifier model are given in Table 3.1 and its execution during a conversation is shown in Figure 3.4.

Model Type	Sequential
Optimizer	SGD (lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
Loss Function	Categorical Cross Entropy
total Params	26,661
Trainable Params	26,661
Non-Trainable Params	0
Num. of Layers	5

Table 3.1 Interviewster Tensorflow Model Details. We used a sequential tensorflow classifier with SGD optimizer and 5 layers

Layer Type	Output Shape	Param
Dense	(None, 128)	16000
Dropout	(None, 128)	0
Dense	(None, 64)	8256
Dropout	(None, 64)	0
Dense	(None, 37)	2405

Table 3.2 (Continued)



```
Candidate Answer:  nasılsın?
After cleaning:  nasıl
1/1 [==============================] - 0s 154ms/step
Returned Intent list:  [{'intent': 'hatır1', 'probability': '0.9981933'}]
Chatbot Answer:  İdare ediyorum. Sizin nasıl gidiyor?
Candidate Answer:  hazır hissetmiyorum
After cleaning:  hazır hissetmek
1/1 [==============================] - 0s 75ms/step
Returned Intent list:  [{'intent': 'başlama red', 'probability': '0.99999857'}]
Chatbot Answer:  Peki. İsterseniz ısınmak için biraz sohbet edebiliriz. Hazır olduğunuzda lütfen 'başlayalım' yazın
```

Figure 3.4 Intent classification examples

3.4. The Decision Component

The decision component assesses the candidate's answers to CBI questions and decides if the candidate has the required competency or not. This component uses transformer technology with Pytorch for assessments and mainly utilizes BERT transformers.

For this component, we used a dataset created by collecting interviews of real candidates and interviewers. The data contains "competency name", "candidate answer" and "interviewers assessment grades" (1 to 5). We added a new feature called "Result" and labeled the candidates as fail and pass (0 and 1). This label is used for making predictions.

3.4.1. The Use of Transformers in NLP

An innovative architecture, the transformer, has been widely used in NLP in recent years. Transformer models tackle sequence-to-sequence problems while skillfully managing long-range dependencies. They do not use sequence-aligned RNNs or convolutions to compute representations of its input and output, but rather depend solely on self-attention (Kulshrestha, 2020). The Transformer model in NLP introduces an 'attention' mechanism that takes into account the relationship between all the words in the sentence. It creates differential weightings indicating which other elements in the sentence are most critical to the interpretation of a problem word. In this way ambiguous elements can be resolved quickly and efficiently (Negri, 2021). For instance, in Figure 3.5, the encoder self-attention distribution for the word "it" from the 5th to the 6th layer of a Transformer trained on English to French translation (one of eight attention heads) is given (Uszkoreit, 2017).

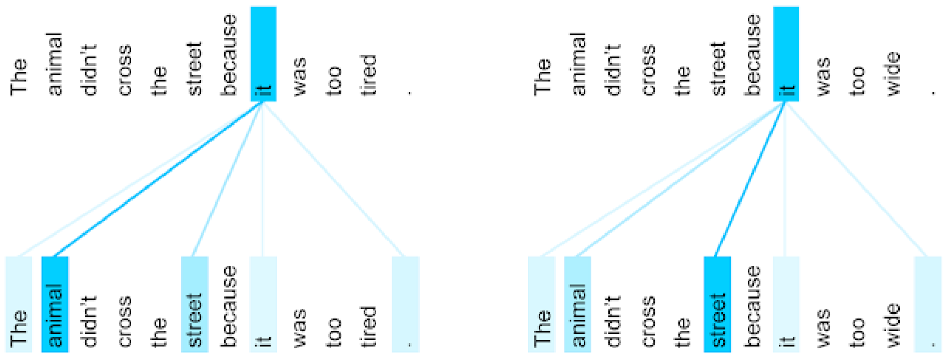


Figure 3.5 Transformer approach. In the first sentence pair "it" refers to the animal, and in the second to the street. While translating these sentences into French or German, the translation for "it" depends on the gender of the noun it refers to. Transformer translates both of these sentences to French correctly *(Uszkoreit, 2017)*

3.4.2. The BERT Model (Bidirectional Encoder Representations from Transformers)

BERT is designed to pre-train deep bidirectional representations from unlabeled text by jointly conditioning on both left and right context in all layers. As a result, the pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, such as question answering and language inference, without substantial task specific architecture modifications (Devlin, Chang, Lee, & Toutanova, 2019). A very instructive 3D Figure of BERT is given in Figure 3.6 (peltarion.com, 2022).

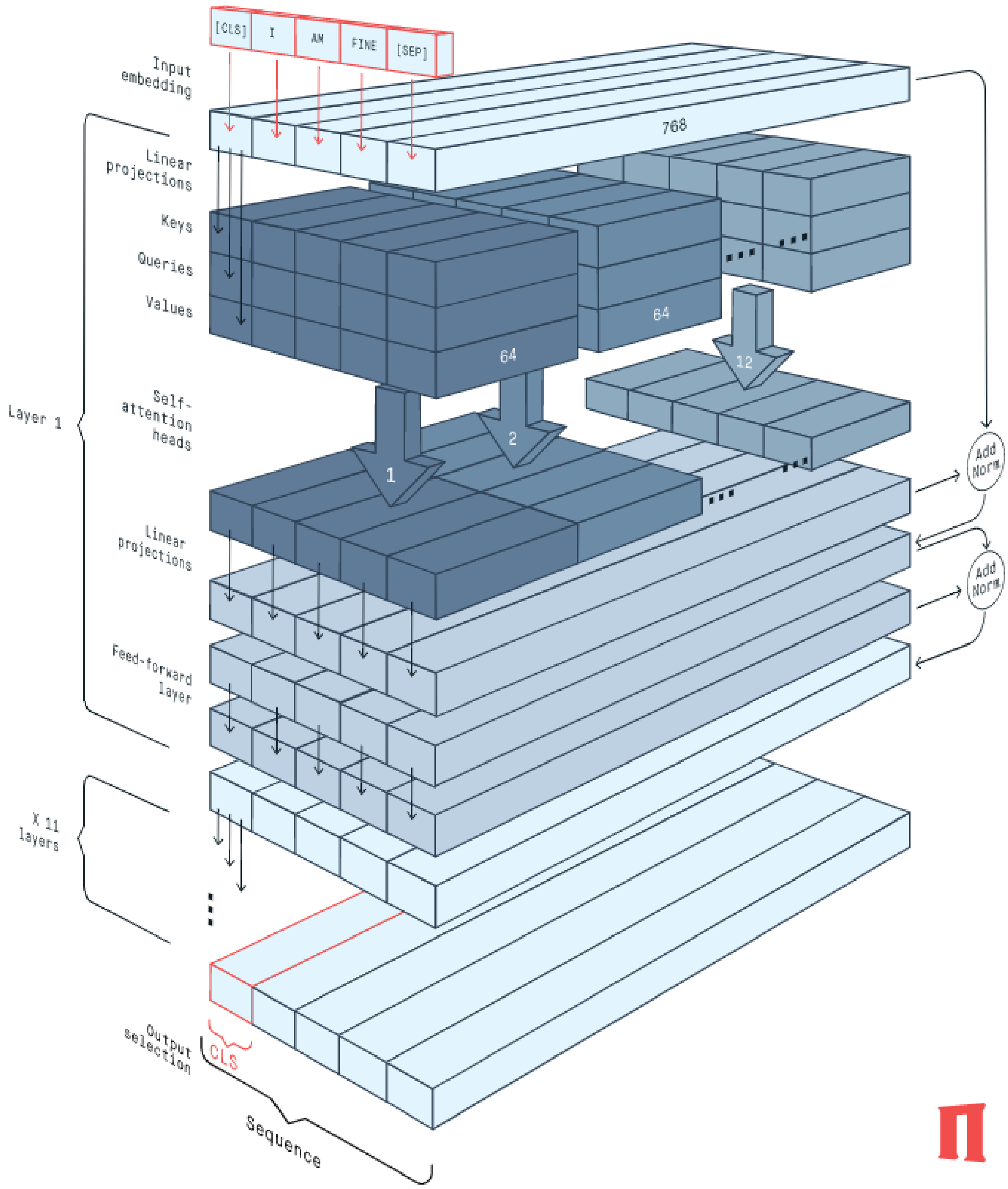


Figure 3.6 BERT model uses 12 layers of transformers block with a hidden size of 768 and number of self-attention heads as 12 and has around 110M trainable parameters

Each token in the block's input is first encoded into a learnt embedding vector that is 768 bytes long. Each embedding vector is then gradually modified each time it passes through a layer of the BERT Encoder. Every embedding vector generates a triplet of 64-long vectors known as the key, query, and value vectors by linear projections. All of the embedding's key, query, and value vectors are processed by a self-attention head, which produces a 64-long vector for every input triplet.

BERT is context-aware because each output vector from the self-attention head is a function of the whole input sequence. Twelve distinct triplets of key, query, and value vectors are produced by a single embedding vector using various linear projections; each triplet undergoes its own self-attention head. As a result, each self-attention head may concentrate on various facets of the tokens' interactions with one another.

In order to take advantage of deep non-linearity, the output from all of the self-attention heads is first concatenated, followed by another linear projection and a feed-forward layer. to improve resilience, residual connections from earlier states are also exploited. A series of altered embedding vectors are produced as a consequence, and they are then processed by the same layer structure 11 more times.

The embedding vectors are altered to include more precise information about each token after the 12th encoding layer. The BERT Encoder block gives the option of having it return all of them or just the first one which is frequently enough for classification tasks (peltarion.com, 2022).

The Interviewster's BERT model "BERTurk"[7] is a taken from Hugging Face website. Datasets given by the Turkish NLP community are utilized for pre-training and assessment. The version we used was trained using the Turkish OSCAR dataset, a recent Wikipedia dump, multiple OPUS corpora, and a unique corpus donated by Kemal Oflazer. The final training corpus has 4,404,976,662 tokens and is 35GB in size (bert-base-turkish-uncased, 2022).

3.4.3. DeBERTa (Decoding-enhanced BERT with disentangled attention)

DeBERTa is a transformer-based neural language model pretrained on large amounts of raw text corpora using self-supervised learning. DeBERTa is intended to learn universal language representations that can be adapted to various downstream natural language understanding tasks. DeBERTa improves previous state-of-the-art pre-trained language models (for example, BERT, RoBERTa, UniLM) using three novel techniques: a disentangled attention mechanism, an enhanced mask decoder, and a virtual adversarial training method for fine-tuning (Pengcheng, Xiaodong, Jianfeng,

[7] https://huggingface.co/dbmdz/bert-base-turkish-uncased

& Weizhu, Microsoft DeBERTa Surpasses Human Performance on the SuperGLUE Benchmark, 2021). In Figure 3.7, the architecture of DeBERTa is given .

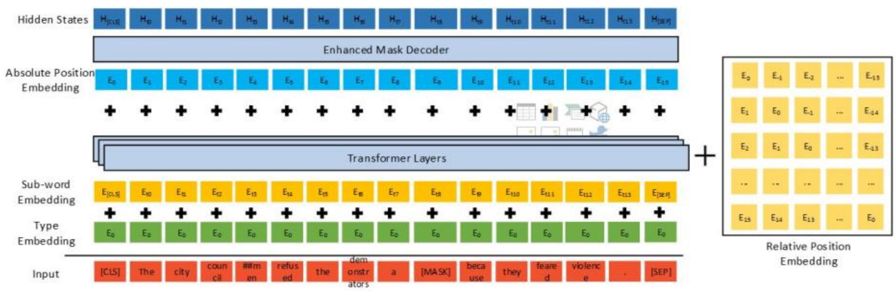


Figure 3.7 The architecture of DeBERTa. DeBERTa improves the BERT and RoBERTa models using a disentangled attention mechanism where each word is represented using two vectors that encode its content and relative position, respectively, and an enhanced mask decoder (Pengcehng et al., 2021)

The model we used in Interviewster is a fine-tuned version of "microsoft/mDeBERTa-v3-base" (a multilingual version of DeBERTa V3) using a reviewed version of well-known Turkish Named Entity Recognition (NER) dataset[8] (mDeBERTa-v3-base-turkish-ner, 2022).

3.4.4. ELECTRA (Efficiently Learning an Encoder that Classifies token Replacements Accurately)

In ELECTRA, four academicians introduced substituted token detection, a pre-training task where the machine learns to differentiate between actual input tokens and convincing but artificially created replacements. Their approach corrupts the input instead of masking it by swapping out certain tokens with samples from a proposed distribution, which is often the result of a tiny masked language model. By using this corruption process, the BERT network's inconsistency in detecting fake tokens during pre-training but not when it is being fine-tuned on subsequent tasks is fixed. After that, they used the network's pre-trained discriminator to forecast whether each token is an original or a replacement (Clark, Minh-Thang, Quoc, & Christopher, 2020). Figure 3.8 presents the Glue Score of ELECTRA obtained in this study.

[8] https://github.com/stefan-it/turkish-bert/files/4558187/nerdata.txt

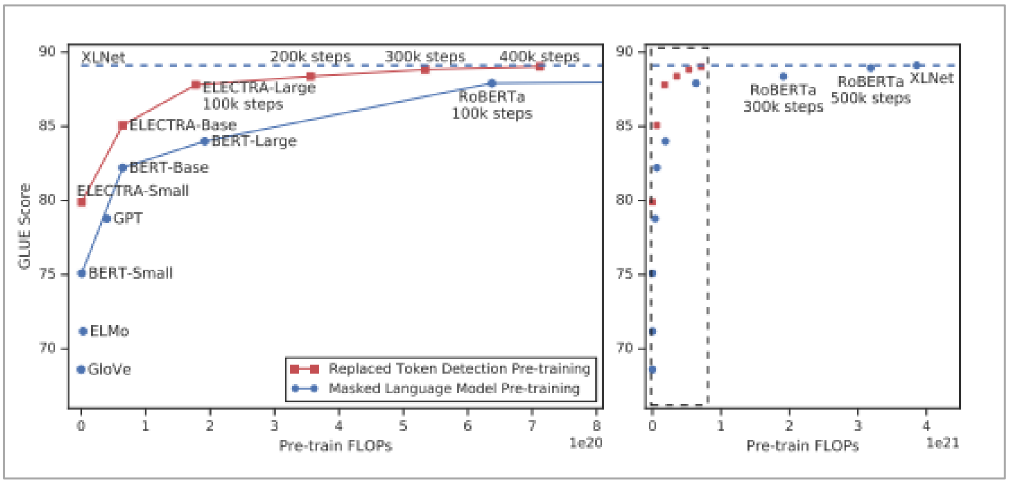


Figure 3.8 Glue Score of ELECTRA. Replaced token-detection pre-training model Electra performed better from masked language models like BERT and RoBERTa (Clark et al., 2020)

A filtered and sentence-segmented version of the Turkish OSCAR corpus, a recent Wikipedia dump, multiple OPUS corpora, and a unique corpus donated by Kemal Oflazer were utilized to train the ELECTRA model that we employed. The final training corpus has 4,404,976,662 tokens and is 35GB in size (electra-base-turkish-cased-discriminator, 2022).

3.4.5. Implementational Details

We used AdamW (Adam with decoupled weight decay) optimizer for our decision model. The extended stochastic gradient descent algorithm known as the Adam optimizer has been used in a variety of deep learning applications, including computer vision and natural language processing. Adam made his debut in 2014. Adam adjusts the learning rate for each neural network weight by estimating the first and second moments of the gradient. The most effective stochastic optimization, Adam, is suggested as needing just first-order gradients in cases where memory is insufficient (Ajagekar, 2021).

But Adam seemed to find fresh life towards the end of 2017. In their article, Ilya Loshchilov and Frank Hutter noted that weight decay appears to be done incorrectly in all libraries using Adam, and they suggested a straightforward correction (which they refer to as AdamW). Despite having rather inconsistent findings, they did present some optimistic graphs, including the one in Figure 3.9 (Loshchilov & Hutter, 2019):

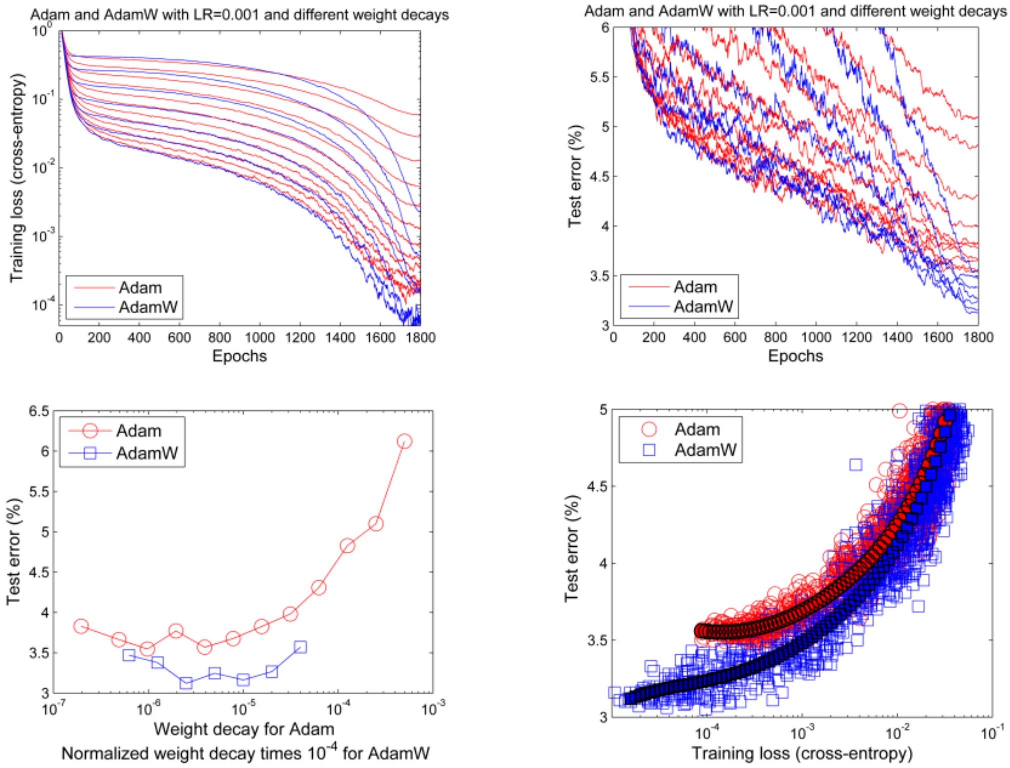


Figure 3.9 Adam vs AdamW. In the graphics below the difference in the training loss, test errors and weight decay between Adam and AdamW is given (Loshchilov & Hutter, 2019)

In our model we used AdamW optimizer with a scheduler for learning rate annealing. Our starting learning rate was **5e-5** for all PML models. We used the "linear schedule with warmup" scheduler which create a schedule with a learning rate that decreases linearly from the first learning rate set to 0 in the optimizer. In this function there is also a warmup period. It increases linearly from 0 to the initial learning rate set in the optimizer. In our model we didn't use the warmup step. The learning rate change in our model is given in Figure 3.10.

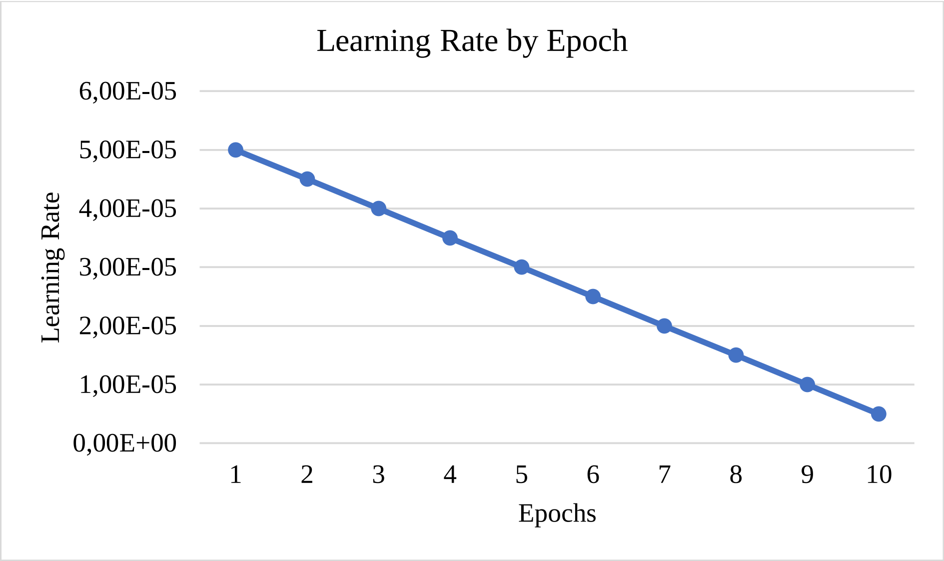


Figure 3.10 Learning Rate Change by Epoch. Our scheduler decreases learning rate in every epoch

4. EXPERIMENTS AND RESULTS

4.1. Dataset Details

While testing the accuracy of **CBI** predictions, we concentrated on four competencies:

- **Learning Capability**

  In this competency, we measure a person's capacity to learn new things and develop himself/herself, based on past learning activities.

  Sample Candidate Answer:

  > I graduated in 2017. I got a computer certificate while I was in school. I went to the course for 6 months. I received body language, diction and personal development certificates. After graduation, I started to work in a call center in a private institution. I was training the new staff there. Later, after working at the branch for 1 year, I left due to marriage. In this process, I received SEGEM certificate. I am currently working as an administrative affairs specialist in a foundation. I took notes on every incoming customer call. I gave training to the new recruits.

- **Customer Focus**

  While measuring this competency, we assess how customer-oriented the candidate can act in the work he/she will do, based on the examples given by the candidate.

  Sample Candidate Answer:

  > Our neighbor loves gardening, he/she was trying to carry things with his wheelbarrow while I was on the phone on the balcony, so I got down and helped him. There was a problem with payment in phone sales. When I realized that a friend of mine would have trouble paying, I offered other options that were more affordable. This made him happier. The busboys who had just started at the hotel had difficulty doing as they were told. They had entered into a negative dialogue with a tourist. I helped my friend, I said that

it would be better to add please at the beginning or end of the sentence. I said don't show your anger to the customer. After a while he/she was then promoted from busboy to waitress.

- **Success**

In this competency, past efforts of the candidate in order to succeed are evaluated and his/her potential to reach the given targets in the future is measured.

Sample Candidate Answer:

I thought that the last company I worked for had deficiencies in marketing. I was examining and sharing the remarkable things about the subject. I was informing the board of directors by following the latest trends in this regard, and also sharing new developments on the websites of rival companies by taking screenshots. After my posts, we took action on sales from the internet, and it was decided to publish an introductory video on youtube. On another issue, there were very rapid price changes in the market, and I conveyed that the sms and e-mail information of the customers should be updated in order to inform the customers. However, it was rejected. A customer came and asked for a product, but the product was not in our stock at that time. In order not to lose it to a competitor, I started to ask questions to the customer. I offered an affordable product that would work for him, he/she accepted and I sold it.

- **Networking**

In this competency, the examples given by the candidate in establishing and using relationships are evaluated, and his/her potential performance in networking and relationship development in the future is predicted.

Sample Candidate Answer:

> After the interview invitation came, my computer broke down, I got a computer from my friends. I was afraid to be alone. A friend suggested I get a pet, so I adopted a cat. It has been very helpful. During my master's period, my teacher and I could not agree on the subject of the thesis. My teacher wanted me to work on something else. On the other hand, I told him that the subject I was working on was very up-to-date and had not been studied much, and I convinced him. I suggested to a friend of mine to become a financial advisor. It suited him too. Currently a trainee financial advisor.

We used a dataset of CBI question answers which are provided by real candidates in response to questions asked by professional interviewers. In standard practices a five-point scale from 1 to 5 is being used. In some practices HR professionals can use non-scalar metrics like 1+, 2-, 2+ to asses candidate more effectively. This 13-point scale was used to create our data as well. However, after several tries, we were unable to successfully estimate the 5th and 13th scales. We worked by simplifying it to a pass/fail scale and putting more attention on the result than the score.

We used %70 of our dataset for training, %15 for validation and %15 for test. Our dataset has three columns named “competency”, “label” and “text”. We have no null values. More dataset details are given in Table 4.1. Train, validation and test distribution of the dataset is given in Figure 4.1.

COLUMN	DESCRIPTION	VALUE LIST	NON-NULL COUNT	DATA TYPE
competency	Defines the related competency of the candidate answer	Learning Success Customer Networking	0	Text
label	Defines if the candidate fails or passes from that competency	Fail Pass	0	Text
text	Candidate's answer to the competency based question	Free-form text	0	Text

Table 4.1 Dataset details

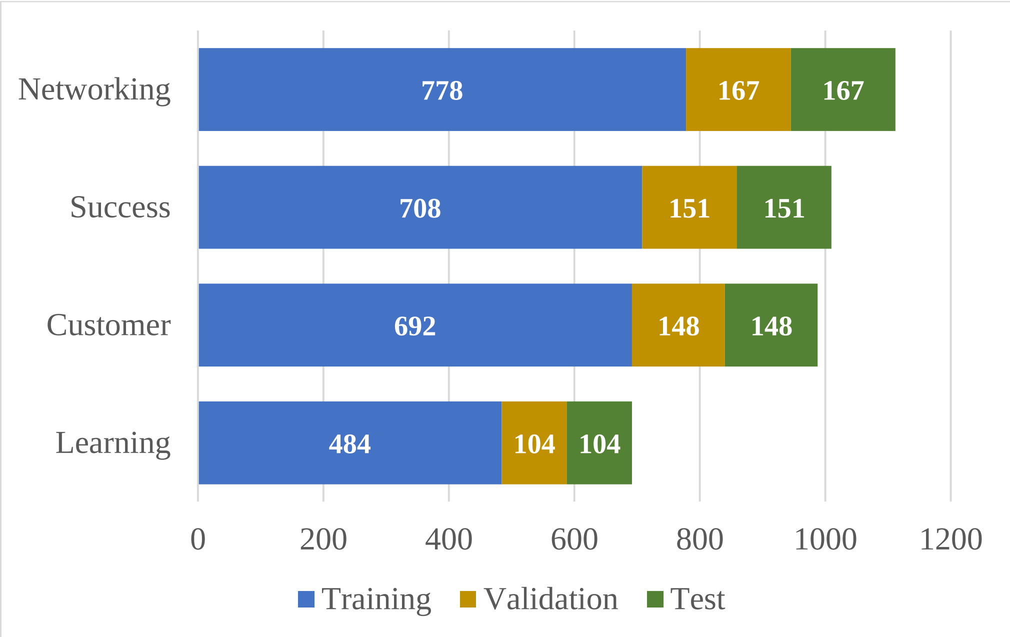


Figure 4.1 Train, validation and test distribution of the dataset

In our initial trials, we have encountered the following issues related to the scale being used:

- When we tried to make an estimation on a five-point scale, the estimation success was measured in the range of 55-60% in all four competencies. When we analyzed the data one by one, we realized that the reason for this low performance was that the interviewers evaluated not only what the candidate said, but also his/her features/behaviors such as the body language and tone of voice, which was impossible to retrieve from written notes.

- The inadequacy of notes written by the interviewers greatly misled the estimation. However, excessive and unnecessary information also disrupted the learning success.
- The grammatical errors caused by taking interview notes quickly also affected learning success. to overcome these errors, correction algorithms should be used and a lot of manual data cleaning should be performed.
- It is very difficult to make an accurate estimation for the candidates who score at the pass/fail border, and the text processing algorithm cannot effectively make correct predictions. Therefore, the records that were very close to the pass/fail border should be eliminated from the dataset.

4.2. Evaluation Results

In our experiments, we used four different models with three being pre-trained models. Each model was trained for 10 epochs and evaluated on the same test data that corresponds to 15% of our dataset.

4.2.1. TF-IDF Model

As a baseline model, we first tried TF_IDF algorithms for all four competencies. In 2, true and false positive and negative predictions (i.e., the confusion matrix) of the TF-IDF model for every competency are given. Table 4.2. presents the performance scores of the model. We observed that the TF_IDF model can produce relatively successful results on our dataset (between 0.76 and 0.80 F1 score) with the highest performance achieved for the competency "success". This might be attributed to the fact that the candidates who provide the competency can give much more examples than those who do not.

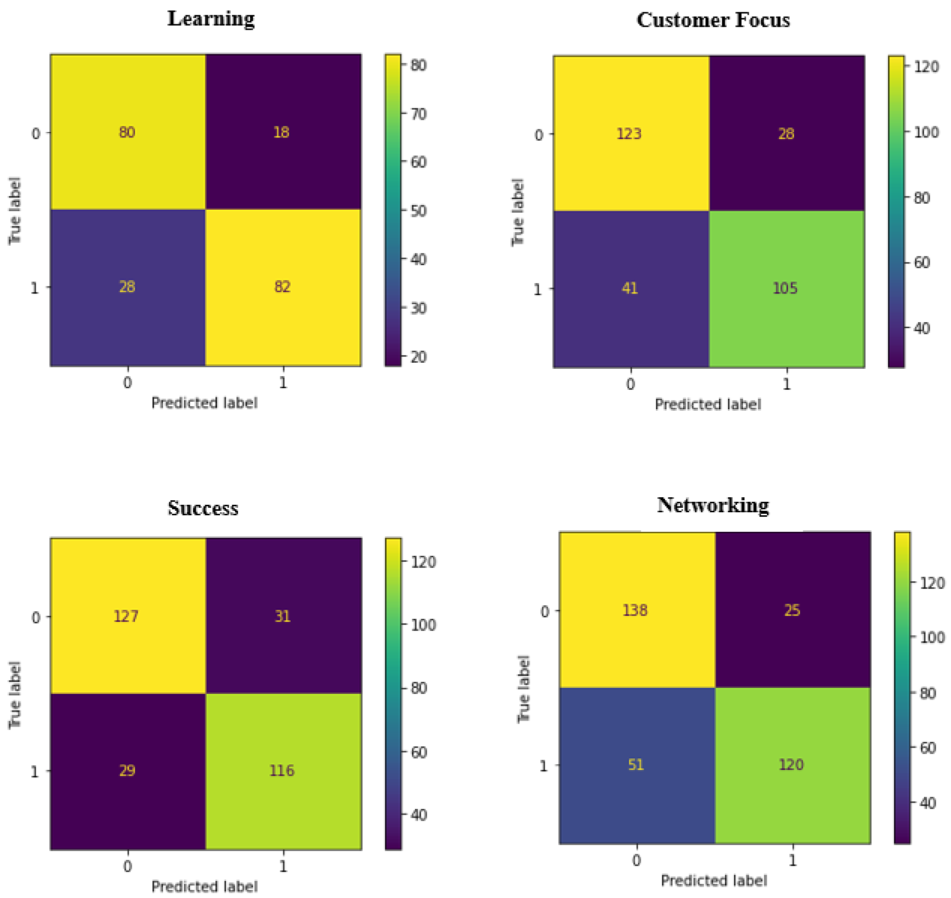


Figure 4.2 The confusion matrices of the TF_IDF model

	Learning	**Customer**	**Success**	**Networking**
F1 Score	0.7788	0.7668	0.8017	0.7717
Precision Score	0.7803	0.7697	0.8016	0.7788
Recall Score	0.7808	0.7668	0.8018	0.7741

Table 4.2 The performance results of the TF_IDF model

4.2.2. Pre-Trained Transformer Models

For each competency, we trained three different models: BERT, DeBERTa and ELECTRA using the same model and the tokenizer. We used the AdamW optimizer for training. As shown in 3, same number of positive and negative examples were used for training the model with respect to each competency. The performance scores were measured using precision, recall, F1 score, and accuracy.

	Learning	Customer	Success	Networking
total Num. of Examp.	692	988	1010	1112
Fail	346	494	505	556
Pass	346	494	505	556

Table 4.3 Training dataset details for each competency

4.2.2.1. Learning Capability

As seen in Table 4.4, the most successful prediction was made by BERT model for this competency. In addition, as shown in Figure 4.3, the training loss values of all three models decreased to values below 0,02 after 10 epochs. It is particularly noteworthy that the BERT model reached 0,01 after 8 epochs.

	BERT	DeBERTa	ELECTRA	TF_IDF	BEST SCORE
F1-score	**0,86**	0,78	0,85	0,78	**0,86**
Training Time*	**1:46:45**	3:29:01	1:30:08	<5 min	**1:46:45**

* h.mm.ss

Table 4.4 F1 scores and training times of all models according to "learning capability"

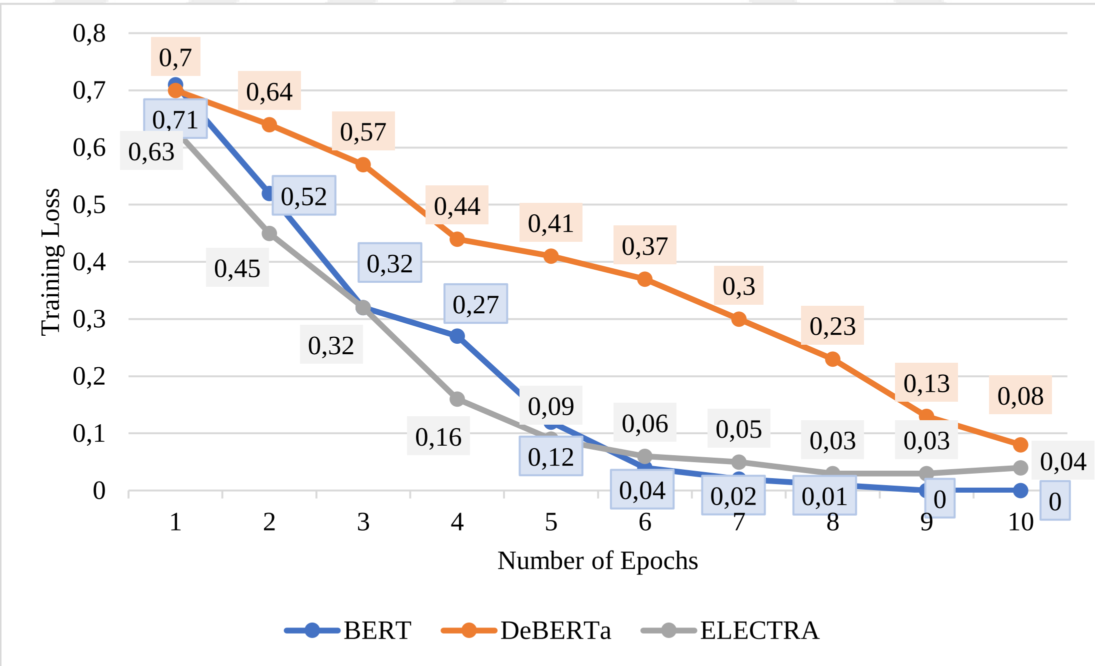


Figure 4.3 Training Loss for "learning capability"

The training and validation loss values are shown in Figures 4.4, 4.5, and 4.6. No signs of overfitting, underfitting or unrepresentativeness were spotted.

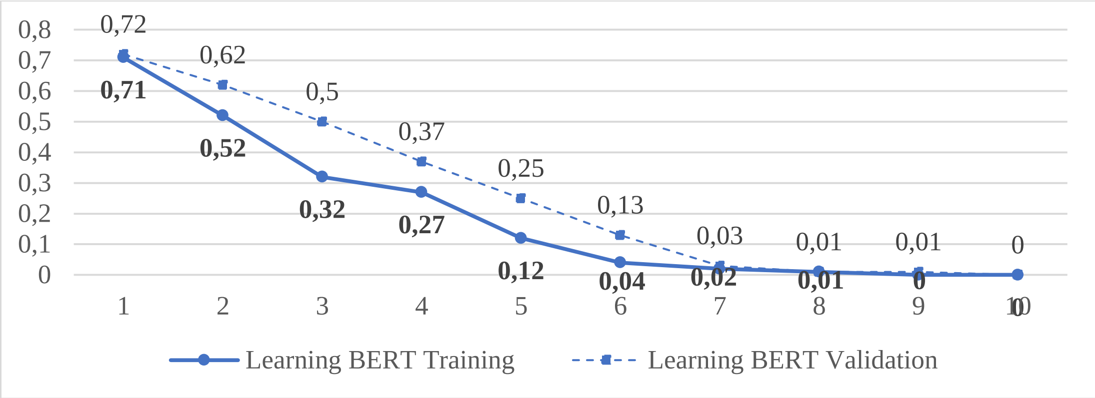


Figure 4.4 Test vs. Validation Training Loss for Learning Capacity with BERT model

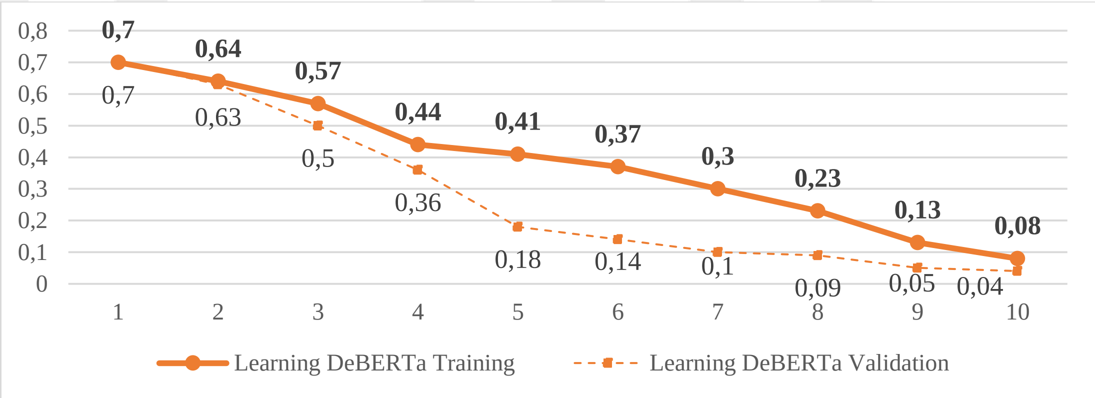


Figure 4.5 Test vs. Validation Training Loss for Learning Capacity with DeBERTa model

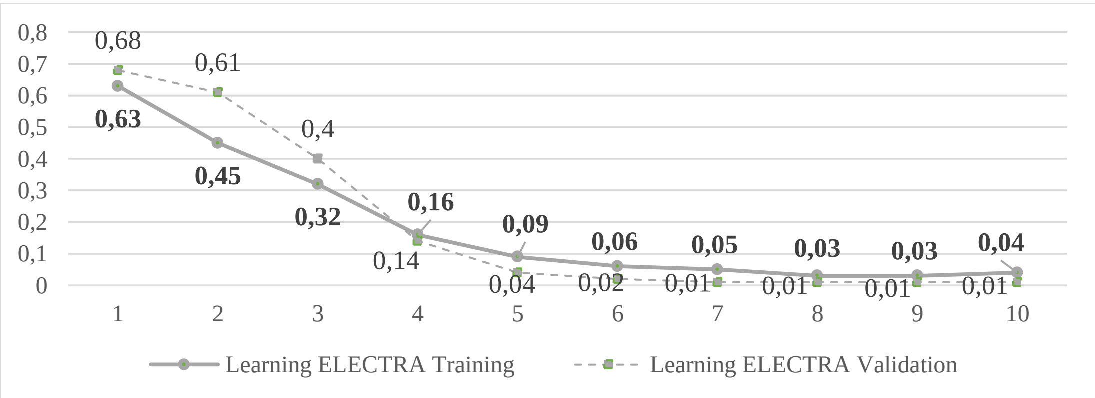


Figure 4.6 Test vs. Validation Training Loss for Learning Capacity with ELECTRA model

BERT	Fail	Pass	Accuracy
Precision	0,80	0,93	0,86
Recall	0,94	0,77	
F1-score	0,87	0,84	

Table 4.5 The performance scores of BERT according to "learning capacity"

DeBERTa	Fail	Pass	Accuracy
Precision	0,84	0,74	0,78
Recall	0,70	0,87	
F1-score	0,76	0,80	

Table 4.6 The performance scores of DeBERTa according to "learning capacity"

ELECTRA	Fail	Pass	Accuracy
Precision	0,83	0,86	0,85
Recall	0,87	0,83	
F1-score	0,85	0,84	

Table 4.7 The performance scores of ELECTRA according to "learning capacity"

4.2.2.2. Customer Focus

The most successful prediction was made by ELECTRA model for this competency, which can be seen in Table 4.8. In addition, as shown in Figure 4.7, the training loss values of all three models decreased to values below 0,03 after 10 epochs. It is particularly noteworthy that the BERT model reached 0,01 after 7 epochs.

	BERT	DeBERTa	ELECTRA	TF_IDF	BEST SCORE
F1-score	0,87	0,81	**0,87**	0,76	**0,87**
Training Time*	2:35:02	4:52:32	**2:31:18**	<5 min	**2:31:18**

* h.mm.ss

Table 4.8 F1 scores and training times of all models according to “customer focus”

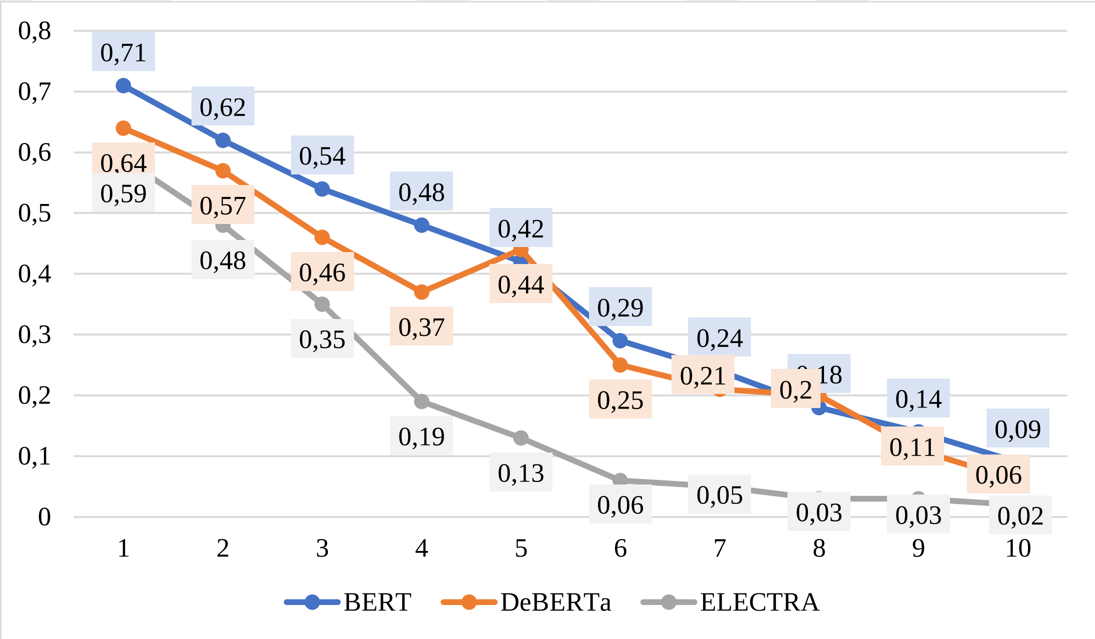


Figure 4.7 Training Loss for "customer focus”

The training and validation loss values are shown in Figures 4.8, 4.9, and 4.10. We did not observe any sign of overfitting, underfitting or unrepresentativeness. On the contrary, values are very close especially in the DeBERTa and ELECTRA models.

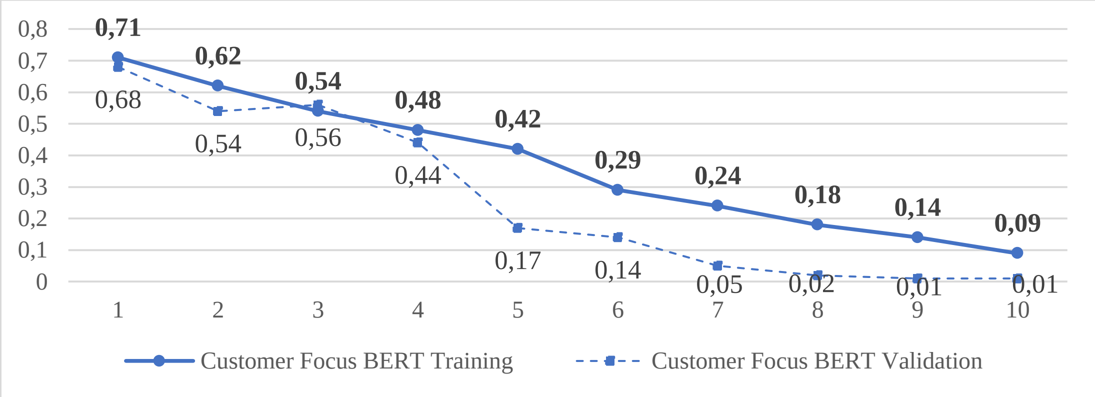


Figure 4.8 Test vs. Validation Training Loss for Customer Focus with BERT model

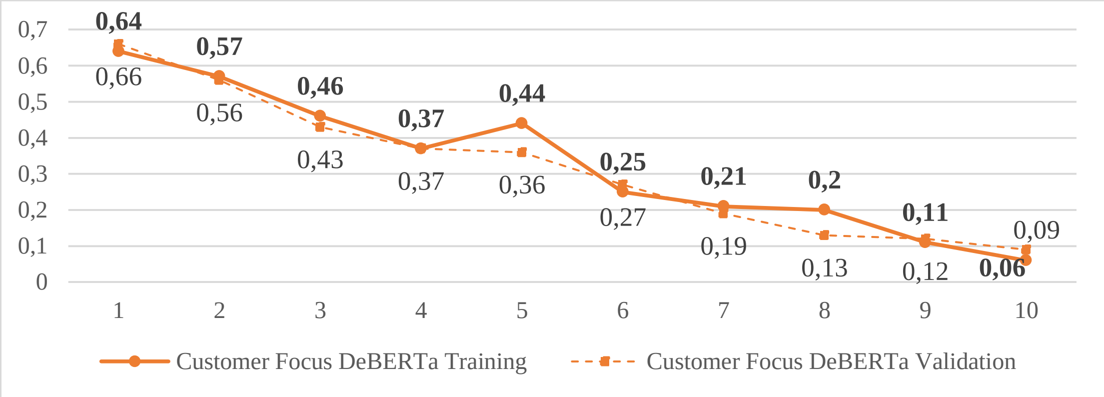


Figure 4.9 Test vs. Validation Training Loss for Customer Focus with DeBERTa model

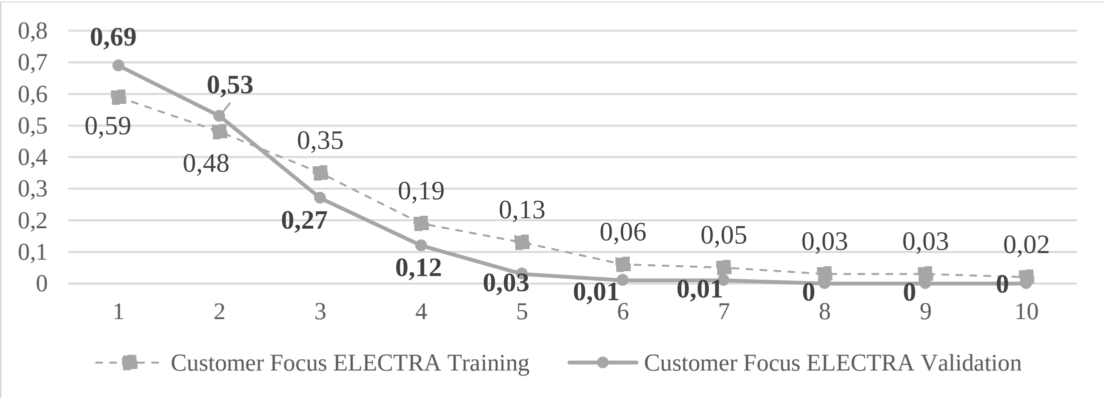


Figure 4.10 Test vs. Validation Training Loss for Customer Focus with ELECTRA model

BERT	Fail	Pass	Accuracy
Precision	0.89	0.85	0,87
Recall	0.84	0.89	
F1-score	0.86	0.87	

Table 4.9 The performance scores of BERT according to "customer focus"

DeBERTa	Fail	Pass	Accuracy
Precision	0,77	0,86	0,81
Recall	0,88	0,75	
F1-score	0,82	0,80	

Table 4.10 The performance scores of DeBERTa according to "customer focus"

ELECTRA	Fail	Pass	Accuracy
Precision	0,84	0,90	0,87
Recall	0,91	0,82	
F1-score	0,87	0,86	

Table 4.11 The performance scores of ELECTRA according to "customer focus"

4.2.2.3. Success

The most successful prediction was made by ELECTRA model for this competency, which can be seen in Table 4.12. In addition, as shown in Figure 4.11, after 8 epochs, training loss values of BERT and ELECTRA models decreased to values below 0,03. However, DeBERTa reached 0,13 after 10 epochs.

	BERT	DeBERTa	ELECTRA	TF_IDF	BEST SCORE
F1-score	0,76	0,80	**0,82**	0,80	**0,82**
Training Time*	2:56:04	6:39:38	**2:13:28**	<5 min	**2:13:28**

* h.mm.ss

Table 4.12 F1 scores and training times of all models according to “success”

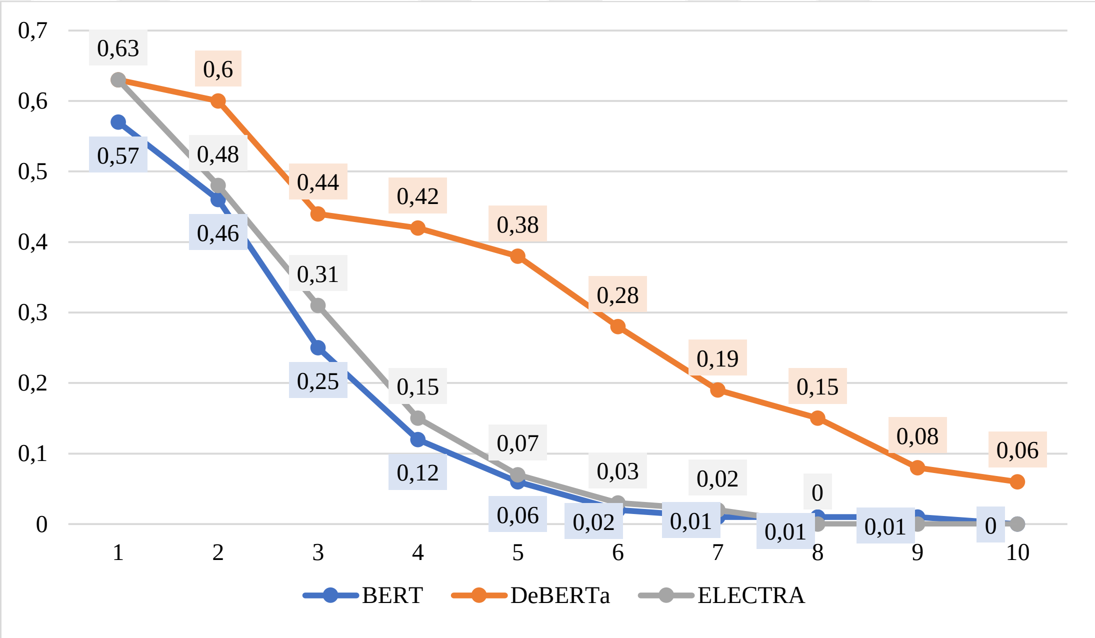


Figure 4.11 Training Loss for "success”

The training and validation loss values are shown in Figures 4.12, 4.13, and 4.14. We did not observe any signs of overfitting, underfitting or unrepresentativeness.

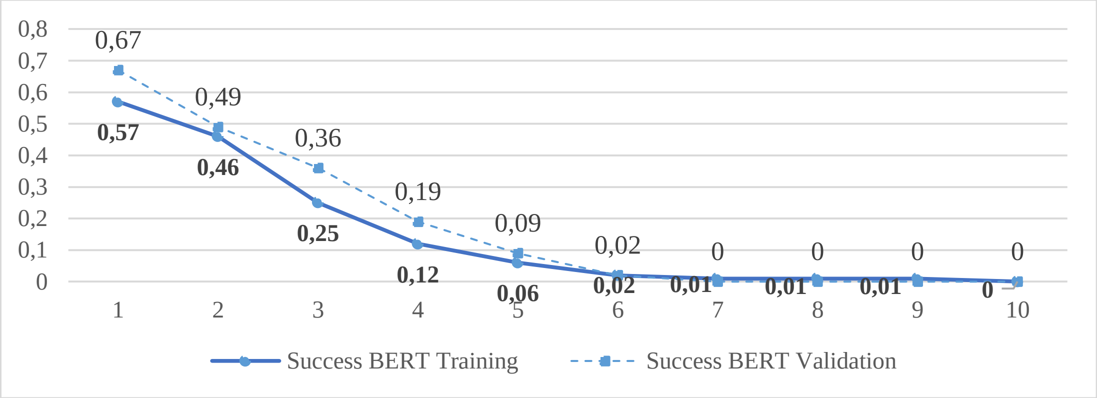


Figure 4.12 Test vs. Validation Training Loss for Success with BERT model

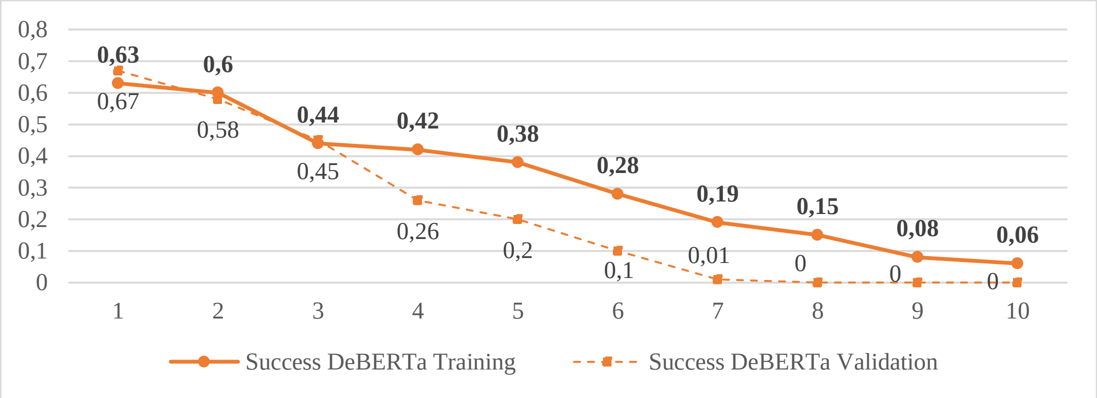


Figure 4.13 Test vs. Validation Training Loss for Success with DeBERTa model

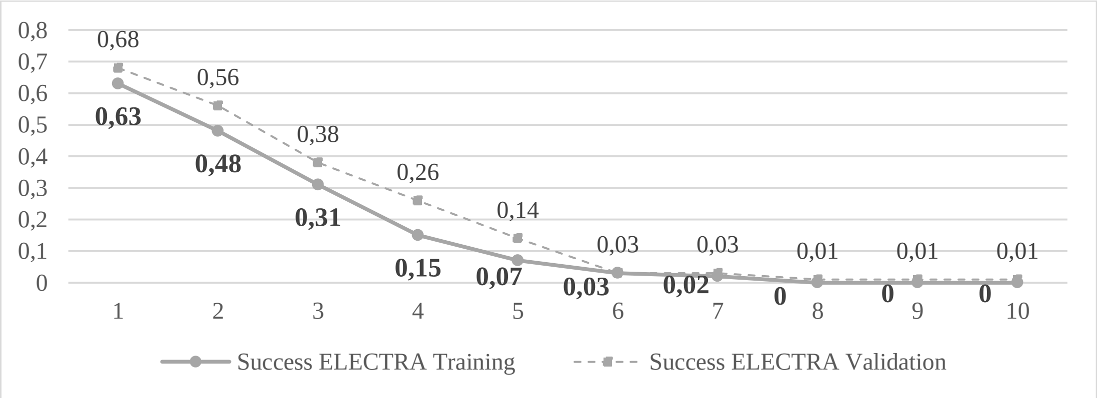


Figure 4.14 Test vs. Validation Training Loss for Success with ELECTRA model

BERT	Fail	Pass	Accuracy
Precision	0,71	0,84	0,76
Recall	0,88	0,63	
F1-score	0,78	0,72	

Table 4.13 The performance scores of BERT according to "success"

DeBERTa	Fail	Pass	Accuracy
Precision	0,75	0,86	0,80
Recall	0,88	0,71	
F1-score	0,81	0,78	

Table 4.14 The performance scores of DeBERTa according to "success"

ELECTRA	Fail	Pass	Accuracy
Precision	0,77	0,90	0,82
Recall	0,92	0,72	
F1-score	0,84	0,80	

Table 4.15 The performance scores of ELECTRA according to "success"

4.2.2.4. Networking

As seen in Table 4.16, the most successful prediction was made by ELECTRA model for this competency. As shown in Figure 4.15, after 10 epochs, training loss values of all three models decreased to values below 0,03. It is noteworthy that BERT and ELECTRA models reached 0 after 7 epochs.

	BERT	DeBERTa	ELECTRA	TF_IDF	HIGHEST SCORE
F1-score	**0,83**	0,73	0,79	0,77	**0,83**
Training Time*	**3:16:05**	6:22:10	2:39:08	<5 min	**3:16:05**

* h.mm.ss

Table 4.16 F1 scores and training times of all models according to "networking"

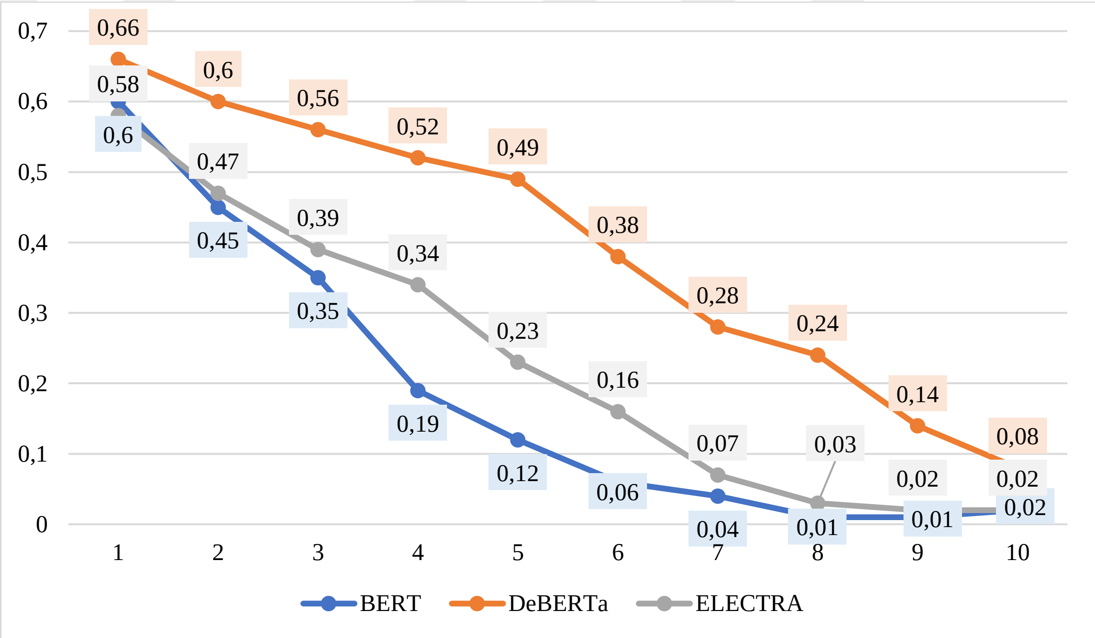


Figure 4.15 Training Loss for "networking"

The training and validation loss values are shown in Figures 4.16, 4.17, and 4.18. No signs of overfitting, underfitting or unrepresentativeness were spotted.

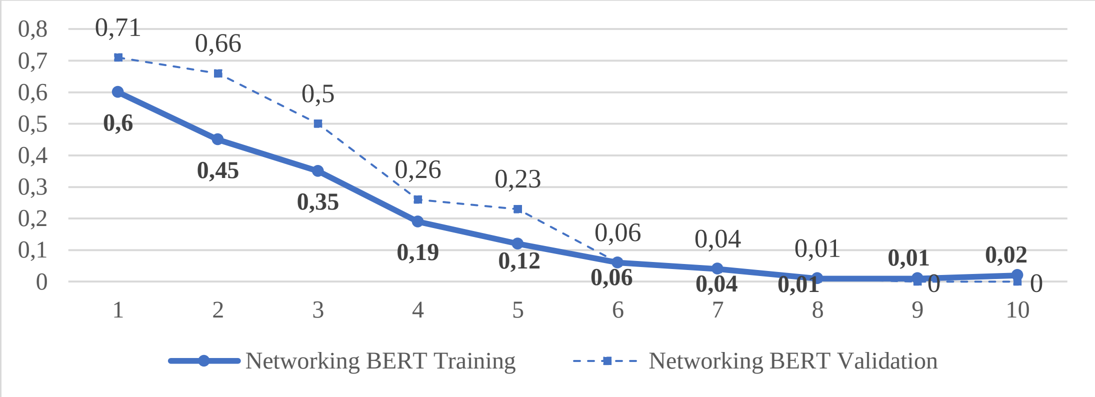


Figure 4.16 Test vs. Validation Training Loss for Networking with BERT model

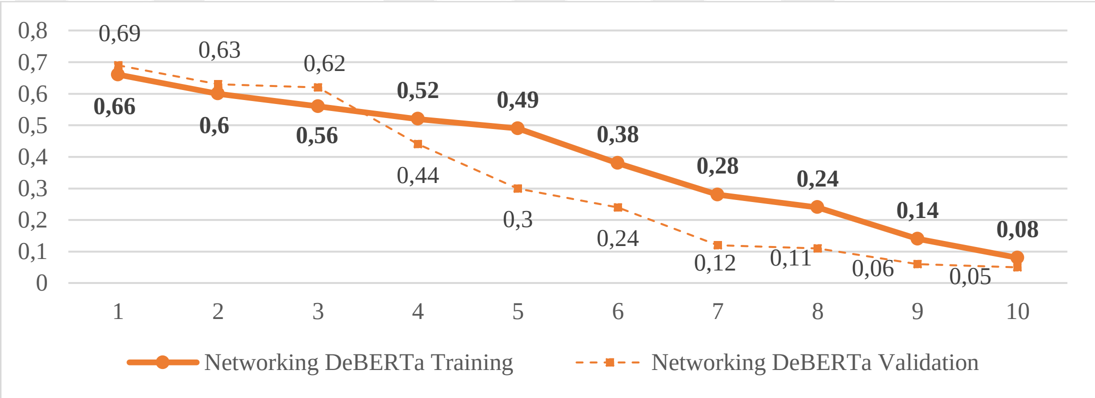


Figure 4.17 Test vs. Validation Training Loss for Networking with DeBERTa model

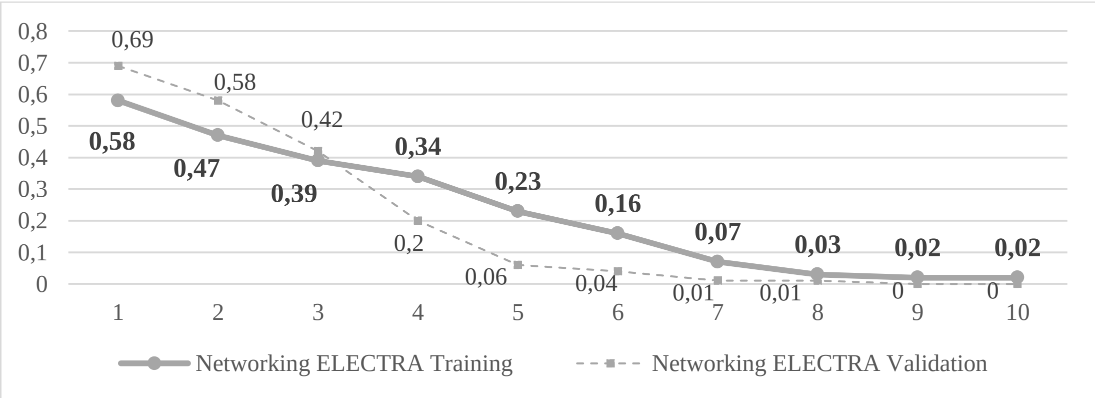


Figure 4.18 Test vs. Validation Training Loss for Networking with DeBERTa model

BERT	Fail	Pass	Accuracy
Precision	0,82	0,85	0,83
Recall	0,86	0,80	
F1-score	0,84	0,83	

Table 4.17 The performance scores of BERT according to "networking"

DeBERTa	Fail	Pass	Accuracy
Precision	0,77	0,70	0,73
Recall	0,65	0,80	
F1-score	0,71	0,75	

Table 4.18 The performance scores of DeBERTa according to "networking"

ELECTRA	Fail	Pass	Accuracy
Precision	0,74	0,87	0,79
Recall	0,89	0,69	
F1-score	0,81	0,77	

Table 4.19 The performance scores of ELECTRA according to "networking"

4.3. Re-assessing False Positive and False Negative Predictions

In order to determine the reasons for false positive and false negative predictions, we ran the model for all data and examined the erroneous predictions. As a test method, we tried to predict the result of the false predicted candidate CBI interview notes manually. We observed that our predictions were almost 40% wrong. Therefore, we understood that the fact that the interviewer notes which are less or more than necessary could mislead the reader. In addition, we noticed that the abbreviations and spelling mistakes that we didn't manage to correct were also included widely in these erroneous predictions.

4.4. General Evaluation of the Results

As we mentioned in Section 3.1, the prediction made by our system would be a common (aggregated) result of three transformer models. After evaluating the model predictions, we found that this approach makes accuracy better. In order to test this, we put the whole dataset into the fine-tuned models and accepted the predictions of at least two of the three models as correct. When we compared the results, we saw that the accuracy values improved slightly as given in Table 4.20.

SUCCESS	BERT	DEBERTA	ELECTRA	COMMON RESULT
ACCURACY	0,86	0,78	0,85	**0,87**

Table 4.20 Common results of three models. The common decision of three models is slightly better than their individual predictions

Our experiments and analysis highlighted the followings:

- With artificial intelligence and especially transformer technology, Turkish proficiency-based interview results can be predicted with a success rate of over 80% as positive (pass) or negative (fail).
- ELECTRA was always the fastest model and DeBERTa was the slowest (more than 4 hours for every model). In Table 4.21 total training times of three models are given.

	BERT	DeBERTa	ELECTRA
total Training Time	10:33:56	21:23:21	8:54:02

Table 4.21 total Training Times of Three Model

- ELECTRA was also the most successful model in our experiments.
- It has been observed, especially during the data cleaning process that a significant portion of erroneous results stem from missing or redundant notes. In this sense, it is thought that the success of estimation can increase with more data and a more detailed and target-oriented data analysis and cleaning process.
- Even the model is successful in an evaluation like pass or fail, the model cannot show success in the evaluations made on a 5 or 13 point scale used in standard HR practices. In order to achieve this, it is thought that it is necessary to increase the size of the dataset and to carry out evaluations in a more professional and controlled form as much as possible.
- At the beginning of the study, we thought that one of the most important achievements of this chatbot would be an unbiased interview evaluation. Although technically this is valid, we conclude that an estimation system working with supervised learning cannot be completely free of bias, but will not go beyond the bias in the sample that creates the dataset, or it will adopt this as an invisible feature. Of course, since we did this study by taking into account the evaluations of more than one interviewer (over 50 in the current study), we can actually accept that this bias is shared at acceptable levels. Therefore, it can be thought that this system provides a certain gain in order to provide an unbiased interview evaluation.
- One of the most important topics here is to understand whether the candidate is telling the truth while giving an example. A significant part of the false positive predictions of the model are rooted from this. If the evaluator did not clearly and meaningfully write that the candidate may not be telling the truth, it’s impossible for the model to catch this. For this, the Interviewster should be able to perform sentiment analysis with image processing technologies and add this to the predictions by extracting text with speech-to-text operations over video responses.

- Another weakness of the program is the necessity of assessing a free text format answer. When interviewing a candidate, candidates are expected to try to polish their experiences. While doing this, they may try different ways such as giving unnecessary details, giving different positive examples that may go beyond the scope of the question and affect the result, and explaining the same example in different sentences. In order to prevent this, different algorithms should be used, and measures such as random human control, keeping track of such behaviors and comparing them with the answers given by the candidates to examine whether there is a similarity and check if any, and perform a human control for all candidates who are evaluated very well or badly can be taken.
- We think that the fact that most of the interviewed candidates are at the beginning of their careers ensures that the examples given by the candidates are similar and therefore the accuracy values increase. If we keep the age range wider (21-30 in the current study), it would be expected that the accuracy values would be lower. However, since such evaluations will be made on a position-based basis, it is expected that a situation with a wider age range, will be rare. Therefore, we don’t think this situation is a major inadequacy.

5. CONCLUSION

Interviewster is a promising application for the human resources technologies industry. It has the potential to become an important tool in the hands of recruiters in order to carry out the candidate evaluation process in a comprehensive, fast, unbiased, and coordinated manner. In addition to being as unbiased as possible, we think that the candidate we mentioned at the beginning of the study will make an important contribution to revealing his/her potential by participating in the interview in his/her comfort zone. It will become a more useful tool when supported by sound and image processing technologies. It can also be functional as an additional application in a career portal.

REFERENCES

Ajagekar, A. (2021, 12 6). Adam. *Cornell University Computational Optimization Textbook*. https://optimization.cbe.cornell.edu/index.php?title=Adam

Akın, A. (2022, 07 05). Zemberek-nlp. *github.com*. https://github.com/ahmetaa/zemberek-nlp

bert-base-turkish-uncased. (2022, 07 02). https://huggingface.co/dbmdz/bert-base-turkish-uncased

Boyatzis, R. (1982). *The Competent Manager: A Model for Effective Performance.* John Wiley & Sons.

Bulat, O. (2022, 02 19). Zeyrek. *github.com.* https://github.com/obulat/zeyrek/

Chatbot Trends Report 2021. (2021, 03 03). chatbotsjournal.com. https://chatbotsjournal.com/chatbot-trends-report-2021-b15479c404e4

Chatbots Explained. (2020, 02 20). ibm.com. https://www.ibm.com/cloud/learn/chatbots-explained

Chrislb. (2022). *ArtificialNeuronModel_english.* Wikimedia. https://commons.wikimedia.org/wiki/File:ArtificialNeuronModel_english.png

Clark, K., Luong, M. T., Le, Q. V., & Manning, C. D. (2020). Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555.*

Devlin, J., Chang, M. W., Lee, K., & toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805.*

drift.com. (2022, 06 08). https://www.drift.com/learn/chatbot/ai-chatbots/

electra-base-turkish-cased-discriminator. (2022, 07 02). https://huggingface.co/dbmdz/electra-base-turkish-cased-discriminator

Ganegedara, T. (2018). *Natural Language Processing with Tensorflow : Teach Language to Machines Using Python's Deep Learning Library*. Pact Publishing.

grabjobs.co. (2022, 07 28). https://grabjobs.co/recruitment-platform/interview-chatbot-guide/

Iuchanka, A. (2022, 03 22). How do chatbots work? Often with a little help from AI. *itechart.com.* https://www.itechart.com/blog/how-do-chatbots-really-work/

Ozen D. & Kizildag, D. (2018). Çalışan Seçim Sürecindeki Görüşmeci Kaynaklı Mülakat Hataları Üzerine Araştırma. *Uluslararası Sosyal Araştırmalar Dergisi. 11*(61). http://dx.doi.org/10.17719/jisr.2018.2988

Kulshrestha, R. (2020, 06 29). Transformers or as I like to call it Attention on Steroids. *towards Data Science*. https://towardsdatascience.com/transformers -89034557de14

Loshchilov, I., & Hutter, F. (2017). Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101.*

mDeBERTa-v3-base-turkish-ner. (2022, 07 02). https://huggingface.co/akdeniz27/ mDeBERTa-v3-base-turkish-ner

nationalcareers.service.gov.uk. (2022, 2 10). https://nationalcareers.service.gov.uk/ careers-advice/interview-advice/the-star-method

Negri, D. (2021, 03 15). Transformer NLP & Machine Learning: size does count, but size isn't everything! *eidosmedia.com.* https://www.eidosmedia.com/blog/ technology/machine-learning-size-isn-t-everything

Öztürk, Ü. (2010). *Performans Yönetimi.* Alfa.

Parvizi, G., & Khishe, M. (2020). Artificial Neural Networks, Concepts, Application and Types. In D. Alexander, *Neural Networks: History and Applications.* Nova.

peltarion.com. (2022, 06 29). https://peltarion.com/knowledge-center/documentation/ modeling-view/build-an-ai-model/blocks/bert-encoder

Pengcheng, H., Xiaodong , L., Jianfeng, G., & Weizhu, C. (2021, 01 6). Microsoft DeBERTa surpasses human performance on the SuperGLUE Benchmark. *Microsoft Research Blog*. https://www.microsoft.com/en-us/research/blog/microsoft-deberta-surpasses-human-performance-on-the-superglue-benchmark/

Purohit, J., Bagwe, A., Mehta, R., Mangaonkar, O., & George, E. (2019). Natural Language Processing based Jaro-The Interviewing Chatbot. 2019 3rd International Conference on Computing Methodologies and Communication (ICCMC), 134-136.

siabot.com. (2022, 07 28). https://siabot.com/industries/enterprise/hr-technical-interview-chatbot

Suakanto, S., Siswanto, J., Febrianti Kusumasari, T., Reza Prasetyo, I., & Hardiyanti, M. (2021). Interview Bot for Improving Human Resource Management. 2021 International Conference on ICT for Smart Society (ICISS), 1-5.

tensorflow.org. (2022, 07 02). https://www.tensorflow.org

Uszkoreit, J. (2017, 9 31). Transformer: A Novel Neural Network Architecture for Language Understanding. *Google AI Blog*. https://ai.googleblog.com/2017/08 /transformer-novel-neural-network.html

Woodruffe, C. (1993). What Is Meant by a Competency? *Leadership & Organization Development Journal*(14), 29-36.