University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Data Science

Reconstructing truthful responses in psychological tests using NLP state-of-the-art models

Supervisor
Prof. Giuseppe Sartori
University of Padova

Co-supervisor
Prof. Mauro Conti
University of Padova

Master Candidate
Roberto Russo

Student ID
2006665

Academic Year
2022-2023

"I suppose therefore that all things I see are illusions; I believe that nothing has ever existed of everything my lying memory tells me. I think I have no senses. I believe that body, shape, extension, motion, location are functions. What is there then that can be taken as true? Perhaps only this one thing, that nothing at all is certain."
— Rene Descartes

Abstract

Psychological questionnaires are a powerful tool to assess psychological conditions such as personality traits, or mental diseases such as depression. They are lists of sentences describing psychological symptoms (e.g. I always think about suicide). The major weakness of this tool is that in many of the scenarios where those questionnaires are used, the subjects who have to respond would benefit from showing a particular condition. Hence encouraging the respondees to answer not according to what they think the truth is, but according to what they think would make them appear the way they believe to be the best in the specific scenario they are into. So to tackle this problem, the first step is to understand when a response is deceiving and the consequent step is to reconstruct what would have been the truthful response. While several machine learning techniques have been successfully applied to detect dishonest responses, no approach I am aware of has been shown to handle the reconstruction of truthful responses properly. Furthermore, all the approaches tried so far both for the first and the second task required a specific model for each type of questionnaire. In this work, I show how can NLP state-of-the-art models be used in a transfer learning framework to address every questionnaire with a unique model, both in lie detection and response reconstruction, obtaining satisfactory results. In this work is also discussed how to improve this approach and further works that can possibly be done.

Contents

Listing of figures

Listing of tables

Listing of acronyms

AI ..... Artificial Intelligence

LLM ..... Large Language Model

ML ..... Machine Learning

SRE ..... Short Random Example

w.r.t. ..... with respect to

1 Introduction

Psychological questionnaires are used to assess a particular aspect of the personality of an individual, such as personality traits, personality disorders, and other kinds of disorders that affect psychological well-being. A psychological questionnaire is composed of questions, called items. Each item is associated with a scale that must be unique for all the items belonging to the same questionnaire, for example, from 1 to 5. For each question the patient has to assign a number from the scale to each item of the questionnaire, depending on how much the patient identifies themself with the item. An item of the questionnaire is generally a sentence such as "I enjoy public events" or "I have been insulted in front of my colleagues", yet "Since a particular event happened I have trouble sleeping at night". Let's assume, for example, that those sentences are items of the same questionnaire that we call "Short Random Example" (from now SRE) and that they appear in the questionnaire in the same order I presented them above. Let's imagine that we choose a scale from 1 to 4 and that one patient has filled out the questionnaire with the numbers (3, 1, 4). With this sequence, the patient is telling us that she/he: enjoys public events, has never been insulted in front of her/his colleagues and that has serious problems every night falling asleep. So, we see that according to the items and the scale, the numbers assume a particular semantic meaning, from which the psychologist can assess some characteristics of the patient. In our example, according to just our three items, a psychologist could say that the patient is an extrovert, that has never been the victim of mobbing, and that the patient has experienced a traumatic event that is causing her/him insomnia. The problem with our SRE is that is too short, and not specialized to assess anything in particular. Even though enjoying public events is generally associated with being an extrovert, it could be the case that our patient likes to go there just to be by herself/himself and talk to nobody. Yet, mobbing does not consist only of being insulted in front of our colleagues, so the patient could never be insulted and still be a victim of mobbing, for example, because she/he is systematically ostracized by everyone at the workplace. Lastly, the difficulty to fall asleep could derive from so many different factors, that only one item about an event after which the patient finds it hard to sleep is simply not enough to determine the source of insomnia. Hence, those questionnaires are usually specialized to assess a particular condition of the patient, and they are made with as many items as needed to accurately pinpoint a specific aspect of

the mental condition of the patient. For example, if we want to assess the personality of the patient, we will have more questions that describe different manifestations of characteristics such as extroversion, eliminating most of the ambiguities. Still for the mobbing, an entire questionnaire is devoted to assessing if the patient has been a victim of mobbing because every item is devoted to a particular type of mobbing. Yet the item about insomnia would be included in a questionnaire that aims to assess the presence of PTSD or any other mental disorder that causes insomnia. The vantage of using the scale is that the answers can be summed and we can choose the items in a way that if the sum is bigger or equal to a certain threshold, called cut-off, we can confidently infer a particular condition. Utilizing illustration, if we devise the questionnaire about mobbing in a way that for each item, the higher the score the more a particular type of mobbing has been done to the patient, then we can set the cut-off such that every patient that surpasses it has surely been affected by systematic mobbing. Those questionnaires have the big advantages that are easy to be used with everyone and allow the psychologist to rapidly evaluate the subject under analysis. The main drawback of this methodology is that an individual, according to the specific circumstances under which the questionnaire is provided, could easily guess which aspect the psychologist is interested in, and manipulate the answer in order to appear in a specific way. This could be the case when a decision that involves the respondee depends on the result of the questionnaire, for example, if hire the subject or not, or if to send a person to prison or not. Therefore, a dishonest individual could manipulate her/his answers to appear extroverted to get a job as a salesperson, even if the candidate is an introvert. Or an accused could manipulate the answers to appear mentally ill when she/he is perfectly conscious and sentient. This main issue could potentially make the whole methodology useless every time the subjects can get advantages if they give a particular response. To address the problem of fake answers, two steps are necessary:

1. Identifying when a fake answer has been given;
2. Retrieving what would be the honest answer given the fake one.

Retrieving the real answer strictly relies on the success of discriminating the real answers from the fake ones because we want to reconstruct only the answers of those who have faked their response. So, one should only focus on the fake answers in trying to find a methodology to retrieve the real answers. For those reasons, from now on, I will discuss those two steps as a separate problem. In the next sections, I will show the unified approach to address the problem of fake answers.

2

What is a lie

To understand this work we need to take a dive into what is a lie from a psychological and cognitive perspective, meaning that we shall go through:

1. How lying has been examined in psychological sciences;
2. Which parts of the brain lying have been proven to involve;
3. How lies affect the usefulness of psychological questionnaires.

2.1 Deception

Deception, or lying, is a central aspect of human behavior and a better understanding of it is relevant in almost all human relationships [1]. In this section, we will first attempt to define the concept of deception, differentiating it from other phenomena that may seem similar but do not fit its explicit definition. We will also discuss the two main ways of lying - dissimulation and falsification - and the factors that influence the frequency and perception of deception. Deception, or lying, is a complex and multifaceted human behavior that involves intentionally stating or transmitting false information in order to mislead others. Unlike fiction, which represents a simulated activity with its own meaning, deception is a conscious and intentional alteration of the truth. Lying can affect both humans and animals [2], and can be intentional or unintentional [2]. It can refer to a type of communication that conveys truly true information to the sender [3]. There are two main ways of lying - dissimulation and falsification. In dissimulation, the liar is hiding or concealing real information without actually saying anything false. Dissimulation is necessary in cases where it is useful to hide emotions. In falsification, on the other hand, the liar is presenting false information as if it were true; the emission of false information through the process of falsification is particularly functional to the main purpose of lying: covering up evidence of what is being hidden from

the recipient. There are various factors that influence the frequency and perception of deception. For example, people are more likely to lie when they have a personal interest in doing so or when they want to protect themselves or others. In addition, the perception of deception can be influenced by our expectations and prejudices toward others. Another factor that can influence deception is the social context in which it occurs. For example, in certain cultural or professional contexts, lying may be more or less acceptable or tolerated. Similarly, certain social norms may encourage or discourage lying. The complexity of this phenom has led researchers to propose various ways of classifying and understanding lying, in order to better understand its causes, consequences, and ways of detecting it. In this section, we review and synthesize existing research on the classification and stages of lying. One way of classifying lies is based on the content of the lie, or the type of information that is being conveyed. [4] proposed a taxonomy of lies that categorizes them based on four types of content: feelings, facts, explanations, and knowledge.

1. Feelings: Lies about feelings involve the communication of false or misleading emotional states, such as pretending to be happy when one is actually angry or sad. These lies can be difficult to detect, as they often rely on verbal and nonverbal cues that are consistent with the false emotional state being conveyed [4];
2. Facts: Lies about facts involve the communication of false or misleading information about events, objects, or people. These lies can be easier to detect, as they often rely on concrete evidence or statements that can be fact-checked [4];
3. Explanations: Lies about explanations involve the communication of false or misleading information about the causes or reasons behind an event or situation. These lies can be difficult to detect, as they often rely on logical reasoning and plausible explanations that can be difficult to challenge [4];
4. Knowledge: Lies about knowledge involve the communication of false or misleading information about one's own or others' beliefs, opinions, or understanding of a topic. These lies can be difficult to detect, as they often rely on subjective interpretation and can be difficult to fact-check [4].

Another way of classifying lies is based on the motivation or purpose behind the lie.[4] proposed a taxonomy of lies based on four types of motivation: self-oriented, other-oriented, principle-oriented, and rule-oriented:

1. Self-oriented lies: These lies are motivated by self-interest or self-preservation, and are meant to protect or enhance the liar's own reputation, status, or well-being. Examples include lying about one's qualifications or achievements in order to secure a job or promotion or lying about one's actions in order to avoid punishment or criticism;
2. Other-oriented lies: These lies are motivated by concern for others, and are meant to protect or benefit someone else. Examples include lying to spare someone's feelings or to prevent conflict or harm;
3. Principle-oriented lies: These lies are motivated by ethical or moral principles, and are meant to uphold or defend a person's values or beliefs. Examples include lying in order to protect someone's privacy or confidentiality, or lying in order to resist injustice or oppression;
4. Rule-oriented lies: These lies are motivated by the desire to conform to social norms or rules, and are meant to avoid social disapproval or to maintain social relationships. Examples include lying to avoid offending someone or to maintain social etiquette.

Lies can also be classified according to the type of lie being told, or the degree to which the lie deviates from the truth. [2] proposed a model of lying that categorizes lies based on three types: white lies, lies of omission, and commission lies:

1. White lies: These are lies that are told with the intention of sparing someone's feelings or avoiding conflict and are typically considered to be minor or harmless. Examples include lying about liking a gift or lying about one's availability to attend an event;
2. Lies of omission: These are lies that are told by withholding or concealing information, rather than by actively communicating false information. These lies can be more difficult to detect, as they rely on the listener assuming that the speaker is being truthful and complete in their communication;
3. Lies of commission: These are lies that are told by actively communicating false or misleading information. These lies can be more easily detected, as they rely on the liar being able to convincingly convey false information.

Another discernment can be made w.r.t. the referent of the lie or the target or object of the lie. [4] proposed a taxonomy of lies based on four types of referent: self, others, situations, and abstract concepts:

1. Self-referential lies: These lies involve the communication of false or misleading information about oneself, such as lying about one's qualifications, abilities, or actions;
2. Other-referential lies: These lies involve the communication of false or misleading information about others, such as lying about someone's intentions, actions, or characteristics;
3. Situation-referential lies: These lies involve the communication of false or misleading information about a specific event or situation, such as lying about the details or outcomes of an event; Abstract concept-referential lies: These lies involve the communication of false or misleading information about abstract concepts or ideas, such as lying about one's beliefs or opinions.

2.2 Deception from a physiological perspective

From a psychological perspective, lying and deception can have significant consequences for individuals and their relationships, and understanding the underlying processes and factors involved in these behaviors can be important for detecting and preventing deception. One influential theory of lying and deception is Paul Ekman and Wallace Friesen's theory of nonverbal cues, which suggests that certain facial expressions, gestures, and vocal patterns may be indicative of deception. This theory has been supported by a number of studies but has also been the subject of criticism and debate. Another important theory in this area is Buller and Burgoon's theory of strategic deception in interpersonal interactions, which proposes that individuals may use deception as a means of achieving their goals in social interactions. This theory highlights the role of context and motivation in deception and suggests that different types of deception may be more or less likely depending on the nature of the interaction. DePaulo and Kashy's theory of the cognitive and emotional processes involved in lying focuses on the internal experience of deception and suggests that lying may involve a variety of cognitive and emotional processes, including memory, attention, and motivation. This theory highlights the complexity of lying and the potential for individuals to experience a range of emotions and thoughts when engaged in deception. Ethical considerations also play a significant role in the study and use of lying and deception. In personal and professional relationships, lying and deception can have serious consequences, and understanding the factors that contribute to these behaviors can be important for preventing and addressing deception. Deception can be approached from two perspectives based

on the role a person assumes during an interaction: one perspective refers to when people intentionally communicate a lie, while the other concerns how people react to a lie [5]. Liars are typically practical people who fulfill their own needs through deception; they often do not consider their lies to be serious, but rather see them as harmless. It is notable that people are often not self-critical when they are the producers of deceptive information; however, this is not the case for their recipients. These recipients tend to adopt a moralistic attitude when considering the lies of others [6]. Another aspect that deserves consideration in the context of deception is the level of motivation present in the people who develop it. Motivation is a determining factor in the success of a lie and inevitably influences the performance and pursuit of the goal, that is, the successful execution of a deceptive performance. Many studies on detecting deception have been criticized because the participants in the experiment were not given sufficient incentives to perform a credible deceptive behavior. Here is a brief summary of experimental methods for detecting deception: these have been divided according to the level of motivation present in the participants [7].

2.3 Deception from a cognitive perspective

The cognitive approach to lying believes that lying is cognitively more stimulating than telling the truth [8], [9]. According to this view, the difference between lying and not lies in the nature of the cognitive processes that support the two distinct communicative intentions. When individuals recount a fact by recounting events that actually happened, they retrieve the story directly from memory: this does not happen during the production of false information, as the lie must be fabricated in the absence of a real historical reference. This new construction of facts is cognitively more demanding than the simple sincere recollection of what happened. To avoid contradicting what has already been said or already known by the recipient, the liar must keep the truth in mind at all times, and at the same time prevent it from escaping his control and emerging while he is presenting the alternative construction of what happened. In addition, the liar must pay attention and monitor both his own behavior and that of his interlocutor, and in seeking to create positive feedback with the latter, he must constantly check the degree of credibility he is achieving or not with the other person [10]. Gombo's theory assumes that deception involves the activation of two cognitive processes: the first functions to control thought mechanisms such as the inhibition of truth; the second, on the other hand, is based on an active management process aimed at analyzing the reactions of the interlocutor, adapting them to one's own attitudes to maintain the deception active and effective [11]. The emphasis on observation and monitoring of behavior is also present within the Interpersonal Deception Theory [12]. This theory is based on a fusion of concepts belonging to interpersonal communication combined accurately with principles belonging to deception detection. According to the authors, deception is structured in the interaction between two or more individuals, based on a tendency to monitor their own behavior. The liar differs from the sincere individual in that the first is performing several tasks simultaneously, inevitably causing a cognitive overload that decreases attentive control of behavior. Therefore, in the first, there is a greater cognitive load that, if not well mastered, could reduce the ability to deceive. This is why the liar must constantly monitor the behavior of the interlocutor and adapt to it, while at the same time controlling the truth, which must not emerge at all costs. The sincere individual, on the other hand, does not have these additional cognitive demands and can simply focus on the truth. The act of lying requires multiple processes in order to allow liars to plan and organize the production of misleading information: the Working Memory Theoretical Model of Deception embraces and values this concept, defining it as necessary for the sender to operate with coherence and lack of

contradiction. According to this theory, the liar must focus on the recipient's reactions while simultaneously modulating their own behavior [13]. There have been numerous scientific publications that have utilized brain imaging during the performance of tasks requiring subjects to produce true or false information. This series of studies confirm the actual increase in cognitive load during tasks of deception. [14] demonstrated how lying involves increased activity in prefrontal regions with concomitant increased reaction times (RTs); in particular, the area that seems to show increased activity is the bilateral ventrolateral prefrontal cortex. As previously mentioned, the prefrontal cortex is the main site of cognitive control, necessary for good masking of the truth. Furthermore, additional brain imaging research has shown that, compared to people producing true information, lying evokes increased and significant activity in brain regions that are typically active during the performance of complex tasks. Conversely, no particular brain regions are systematically activated when subjects are faced with situations or tasks where the veracity of facts is required [9]. It is therefore evident that there is empirical support for the fact that lying has a significantly higher cost compared to the production of truth. Firstly, it should be noted that people themselves define lying as a significantly more demanding process compared to so-called truth-telling [15]. This increase in cognitive load also impacts a more basic level of communication: as cognitive effort increases, there is an increase in the presence of verbal, nonverbal, and paraverbal (voice tone, phrase intonation, etc.) indicators that fall among the most renowned indicators of lying [16]. When asked a question about a fact, this automatically and unintentionally activates the truth stored in long-term memory. In order for the liar to gain the predetermined benefits of deceptive communication, the true information must be quickly inhibited.

3

How Machine learning can be used in our case

Once a comprehension of the functioning of psychological questionnaires has been achieved, it is possible to consider the application of artificial intelligence (AI), specifically machine learning (ML), to address related problems. More specifically, when implementing an AI solution, ML is employed when the problem lacks a mathematical formalization, but it is possible to provide examples of the problem. In such cases, it is possible to transform the original problem into a form that is amenable to a solution using an ML algorithm, taking advantage of the large availability of samples. By training an ML algorithm using many samples, it is possible to formalize the problem so that given specific input, the output can be expected to reflect the correct answer to the problem. The lie detection problem can be viewed as a classification problem, in which the classes are Honest (H) and Dishonest (D), while answer retrieval can be seen as a multi-output regression problem in which the response variable is the scale number that subjects would have assigned if their answer were honest. To train an ML algorithm, the input must be expressed in numerical form. Therefore, it is necessary to translate the data, i.e. samples, into numbers, preserving the meaning of the data. When using the algorithm on new data, this data must also be coded in numerical form in the same manner as the data used to train the algorithm. In the case of psychological questionnaires, the data is already in numerical form, with each answer represented as a sequence of numbers. The only non-numerical feature is the label, which can be transformed into numerical form by assigning $H = 0$ and $D = 1$. It is important to note that there is a distinction between the dataset for the classification problem and that for the regression problem. In the case of the classification problem, the dataset consists of the answers and the label assigned to each of them, while in the regression problem, the dataset comprises the fake answers and the corresponding honest answer. In both datasets, the data is represented as a matrix A, with each row corresponding to a subject and each column corresponding to an item. The last column of the matrix corresponds to the label in the classification problem, while for the regression problem, there are two columns for each item, one with the fake answer and

one with the real answer, with no column for the label. In both datasets, the element a_{ij} of A corresponds to the answer given by subject i to item j.

3.1 The limitations of classic Machine Learning approaches

The first issue in finding a unique solution pertains to the fact that for each type of questionnaire, the n columns correspond to different items, which may be expressed using different scales. Consequently, answer 3 to question j may hold a completely distinct meaning for two questionnaires. Therefore, in order to train a unique algorithm, it is essential to modify the data to account for this difference. Thus far, proposed solutions for both the classification problem and regression problem have addressed the task for only one type questionnaire. In other words, every algorithm has been developed to solve the problem for a single type questionnaire. Furthermore, the reconstruction of the honest answer remains unsolved, as all the proposed solutions to date have not demonstrated sufficient accuracy.

3.2 Overcoming the issues with Natural Language Processing: new approach

As previously expounded in the introduction, each number assigned to a specific item on the scale corresponds to a particular sentence. The answer is a sequence of numbers allocated to a series of items, forming a text with a specific meaning. However, machine learning algorithms necessitate numerical inputs to be processed. Why, then, are we contemplating transforming numbers into text? One of the reasons is simple: if we translate the numerical sequence into sentences, we could manage all the questionnaires together since the words would possess the same meaning, regardless of the questionnaire. Hence, by translating the numerical sequence into sentences and then tokenizing the sentences using the same criterion, we can convert the text back into numbers such that each numerical sequence reflects the sentence's original meaning, irrespective of the questionnaire and its corresponding scale. Consequently, we can use a single algorithm, thereby solving all the questionnaires. However, this reason alone is not sufficient since alternative approaches may encode each answer sequence more rapidly without necessitating the translation into text. Transforming numerical sequences into text confers a significant advantage as it enables us to exploit the potential of state-of-the-art models in natural language processing (NLP). One of the main classification tasks in NLP is sentiment analysis, i.e., the inference of a text's sentiment. State-of-the-art models like BERT models have demonstrated remarkable proficiency in sentiment analysis. This implies that these models can comprehend the semantic meaning of a text and the underlying connections between words that reveal whether a sentence expresses positive or negative sentiments. Therefore, sentiment analysis shares many characteristics with our classification problem, which requires us to detect lies by comprehending the underlying relationship between sentences. Consequently, by transforming our data into text, we can leverage the capabilities of state-of-the-art models for sentiment analysis. For answer retrieval, we can use state-of-the-art text-to-text

generative models by employing the same logic as before. These models are capable of translating one text into another, which we can utilize to "translate" the fake answer into the truthful one. We can achieve this through fine-tuning, which is a type of transfer learning. In later chapters, I will introduce and explain why this approach works. The intuitive idea is to utilize a model that has been trained on task A to complete another task B that shares many features with task A. For example, if a soccer player has been trained for the central task of running, we could expect that the player would be a good runner with a little specialized training. It is reasonable to assume that if two tasks are similar, then a model trained on one of the tasks would be effective for the other with minimal additional training. The last reason supporting the use of NLP for our problem is the use of NLP in problems such as detecting mental diseases from non-clinical text or detecting neurodegenerative diseases from text or speech [17], [18] and [19]. Previous studies have extensively argued that lying involves different parts of the brain than telling the truth and gives rise to more tasks for the brain. Since NLP models have shown the ability to detect mental and brain conditions, such as emotion, mental illnesses like depression, and neurodegenerative disease, from natural language (written or spoken), it is sensible to expect that NLP models can detect lies. Therefore, it is worth exploring the use of these models in the context of answer retrieval.

3.3 The promise of NLP models over traditional ML models

This work aims to investigate the potential of Natural Language Processing (NLP) models in the domains of lie detection and answer retrieval. NLP models are regarded as highly promising compared to traditional Machine Learning (ML) approaches for two key reasons. Once a fine-tuned NLP model has demonstrated strong performance, it is reasonable to assume that it will generalize well to unseen questionnaires and items, a task that is not feasible with traditional ML techniques. This is due to the fact that traditional ML models rely on the explicit encoding of answers to distinguish between items in the training set, and are thus unable to handle input data from items not present in the training data without producing inaccurate predictions. As such, this work has the potential to help paving the way for the development of a novel "truth's machine" based on text, enabling the creation of new methodologies for assessing psychological conditions. This would entail leveraging AI to assist psychologists in designing questions that enable the detection of lies and retrieval of the honest answer, at least at the semantic level. Although the challenge of interpretability arises in this context, it can be addressed by exploring the reasons why certain questions elicit responses that are easier to classify as honest or dishonest than others. This can be investigated by psychologists from a human perspective, thereby allowing for the discovery of new insights with the help of AI.

4

Dataset

The dataset is composed of several types o questionnaires, let's review them briefly:

1. The Negative Acts Questionnaire–Revised (NAQ-R) [20] : aims to assess is one has been a victim of mobbing, 356 participants, 23 items;
2. The International Adjustment Disorder Questionnaire (IADQ) [21]: tests the presence of adjustment disorder, 255 participants, 10 items;
3. Short PID5 [22]: it aims to identify mental disorders, 519 participants 25 items;
4. PCL [23]: aims to identify PTSD, 201 participants 20 items;
5. QD-CBA [24]: assess if the subject is depressed, 314 honests and 321 dishonests. 23 items;
6. IES-R [25]: assess the presence of PTSD, 179 participants, 22 items;
7. PRMQ [26]: assess the memory participants of the subject, 702 participants, 16 items;
8. PHQ9-GAD7, : assess the presence of Anxious-depressive syndrome, is the combination of two questionnaires: PHQ9 [27] and GAD7 [28], 559 participants, 16 items.

For each questionnaire, with the exception of the QD-CBA, participants were administered the questionnaire twice. In the first instance, participants were instructed to respond honestly. In the second instance, participants were instructed to provide exaggerated responses in an effort to simulate the presence of a disorder, a practice commonly known as "Faking Bad". For some of the questionnaires, a measure based on the total scoring was designed to help the diagnostic. This measure is called cut-off. The formula to perform the calculation depends on the specific questionnaire, but all of those formulas involve the sum of all the scores assigned to each item. Not all the types of questionnaires were paired with a specific formula for the cut-off, thus, for those questionnaires that have been associated with a cut-off the reader is encouraged to read [20], [22], [23], [25], [26], [28] to see how it is calculated. For the rest of the questionnaires, the cut-off is the average of the sum of the scores of the honest responses. Participants of the CBA questionnaire were directed to provide either honest or simulated responses,

but not both. As a result, the CBA questionnaire was excluded from the dataset used for the answer retrieval task, as the corresponding honest answers were not available for the simulated responses. The dataset for classification was constructed by including the number of samples equivalent to twice the total number of participants across all experiments. In contrast, for the answer retrieval task, the dataset was composed of samples equal to the total number of participants across all experiments. For each questionnaire, the participants, when they were to give dishonest responses, were instructed to exaggerate the specific mental disorder that the corresponding questionnaire aims to assess. This means that every participant was trying to manipulate their response in order to show evident signs of the presence of a particular mental disease.

4.1 Translating the answers into sentences

Every number in each sample was converted into a corresponding sentence based on the associated item and scale. Given that the participants were all Italian and the questionnaire was administered in Italian, the sentences were written in Italian. To illustrate the process of conversion, we can revisit the example provided in the introduction. Specifically, for the SRE questionnaire, the items were:

1. "I enjoy public events"
2. "I have been insulted in front of my colleagues"
3. "After a particular event I have trouble sleeping at night"

Imagining that the scale employed in our example ranged from 1 to 4 with a unit increment, the conversion would be similar to what is presented below. For item number 1:

1. "I do not enjoy all public events";
2. "I barely enjoy public events";
3. "I enjoy public events";
4. "I really enjoy public events";

For item number 2:

1. "I have never been insulted in front of my colleagues";
2. "I have been rarely insulted in front of my colleagues";
3. "Sometimes I have been insulted in front of colleagues";
4. "I have often been insulted in front of colleagues".

For item number 3:

1. "I don't have trouble sleeping at night since a particular event happened";
2. "I rarely have trouble sleeping at night since a particular event happened";
3. " Sometimes I have trouble sleeping at night since a particular event happened";
4. "I often have trouble sleeping at night since a particular event happened".

Figure 4.1: Word's cloud of honest responses

This should make clear the criterion of conversion. Recalling the answer our hypothetical subject gave in our example (3,1,4), our sentence (so our sample) would be "I enjoy public events. I have never been insulted in front of my colleagues. I often have trouble sleeping at night after a particular event happened." Even though it is not so natural, still as we will see, the models perform well.

4.2 Exploratory data analysis

In order to explore the dataset created in this study, a text exploratory analysis technique known as "cloud of words" was employed. This technique involves the creation of an image in which words are displayed with a size proportional to their relative frequency within the corpus of text. Specifically, the more frequently a word appears in a text or set of texts, the larger it will appear in the image. However, since many words within our dataset do not rely on the subjects who provided the responses, they appear with equal frequency in both honest and dishonest responses. As such, these words are not informative for distinguishing between honest and dishonest responses based solely on their frequency of occurrence and were thus excluded from the Cloud of Words. Additionally, stop words in the Italian language were also excluded from the analysis, as they lack any intrinsic meaning and do not provide informative content for the scope of this exploration. Following this procedure, two separate clouds of words were generated: one for honest responses and one for dishonest responses. In the image presented, denoted as the "honests' cloud" in green and the "dishonests' cloud" in red, it is observed that the latter displays more prominent words such as "often" and "always". This is coherent with the fact that the participants of all the experiments were instructed to exaggerate their responses in order to simulate a particular mental disorder, and the majority of the items in the questionnaires were negative sentences. The occurrence of these temporal adverbs

Figure 4.2: Word's cloud of dishonest responses

emphasizes the frequency of negative events, hence indicating dishonesty. In contrast, the "honests' cloud" shows the most present words to be "rarely" and "never". This opposition between the two clouds suggests that in this particular scenario, higher responses on the scale are generally associated with dishonesty, whereas lower responses are associated with honesty.

5 Models

5.1 Transformer architecture

The present discourse aims to explicate the fundamental component of the models employed for both classification and reconstruction, namely, the Transformer architecture. The Transformer model was initially presented by [29] as a potential alternative to recurrent neural networks (RNNs), which were the prevalent architecture for natural language processing (NLP) tasks during that time. One of the critical limitations of RNNs pertains to their sequential nature, which impairs their ability to process lengthy sequences of input effectively. In contrast, the Transformer model overcomes this limitation through its utilization of a self-attention mechanism, which permits the model to consider all elements of the input sequence simultaneously. To grasp the concept of self-attention, it is essential to first comprehend the attention mechanism.

5.1.1 Attention mechanism

The Attention mechanism serves the purpose of accentuating the resemblance between two objects (here referred to as "items"), which should result in identical values [30]. The principal constituents of the Attention mechanism are the query, key, and value. Each key corresponds to a value, while the query is a request to access a specific value. Unlike the key, there is no specific rule to assign a query to a value. Hence, it becomes necessary to develop a methodology for retrieving the value given the query. The Attention mechanism addresses this need by assuming that the higher the similarity between the query and a key, the greater the likelihood that the value assigned to that key will be assigned to the query. To facilitate comprehension, let us consider the example of navigating through YouTube. While searching for a video, we typically enter specific words, i.e., the query, that we expect to be in the title of the video, i.e., the key, or to be related to the video in some manner, i.e., the value. Once we input the query, YouTube generates a list of videos whose keys correspond most closely to the query. The Attention mechanism

assigns greater similarity to pairs of query-key that are more alike, and hence "pays more attention" to these pairs. Consequently, it utilizes these similarities as weights to determine the average of the values paired with the keys. Note that the attention is calculated for each query, hence a query is compared to a set of keys and their associated values. The mathematical and most general formulation of the attention mechanism is:

$$Attention(q, D) = \sum_{i=1}^{n} \alpha(q, k_i) v_i \tag{5.1}$$

In this context, we consider D as the set of pairs consisting of key and value, and alpha as a similarity measure [30]. A crucial constraint in our weighted averaging approach is that the sum of weights must be equal to 1. As Transformer models are a deep learning technique, the similarity measure is learned from the data rather than being predefined. Specifically, the attention mechanism within a Transformer model learns a family of functions, including the softmax function, which is the multinomial version of the sigmoid function. The softmax function enables computation of the probability that a given query is related to a specific key-value pair. Therefore, learning this function is critical in accurately estimating this probability. So in our context, the Attention mechanism is

$$softmax(\frac{QK^T}{\sqrt{d}})V \tag{5.2}$$

To facilitate the learning of the softmax function, a set of weights is assigned to the query, key, and value, which are learned by the network through matrix multiplication. The Transformer architecture employs matrices as weights for these three components. Notably, the query, key, and value in Transformer all have the same dimension, d. Assuming that the query and key elements are independent and identically distributed with zero mean and unit variance, their dot product will have a zero mean and variance equal to d. To maintain the magnitude of the dot product within reasonable limits, it is scaled by its standard deviation to maintain a unit variance [30]. This method ensures that the dot product remains under control and is conducive to effective training of the model.

5.1.2 Self-Attention

Self-attention is a specific instance of the attention mechanism wherein the query, key, and value all refer to the same object. This approach is known as self-attention due to the computation of similarity between different elements of the same object [30]. In essence, it treats the query as if it is referring to itself and seeks to associate it with the most similar key, which is also part of the query, to obtain the corresponding value. It is worth noting that this application of attention is not meant to retrieve a specific value based on a given key but instead aims to capture the correlations among all the elements of a given object. The usefulness of self-attention is evident in natural language processing tasks, where understanding the context-dependent meanings of words is crucial for interpreting sentences correctly. By employing self-attention, a model can effectively capture the contextual dependencies among the words and phrases in a sentence. For instance, in the sentence "Luke does not enjoy playing guitar as much as he does playing football," the word "playing" has different meanings depending on its association with other words in the sentence. The self-attention mechanism can effectively capture these context-dependent relationships, and the resulting features can aid in the model's comprehension of the sentence's meaning. Additionally, the word "he" is closely related to "Luke" in this sentence, and the self-attention mechanism can account

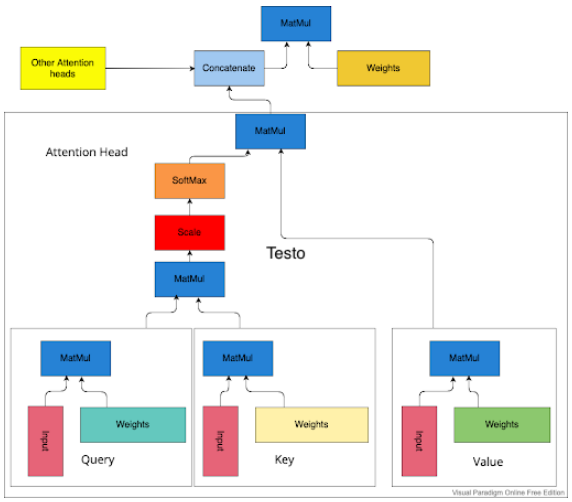

Figure 5.1: Multi-head block.

for this association to ensure that the model correctly identifies the referent.

5.1.3 Multi-head attention

Drawing upon the aforementioned examples, we have observed how words within a sentence are interconnected, and the meaning of a sentence emerges from those intricate connections. We have also discerned that different pairs of words have varying types of relationships, such as (*playing*, *guitar*), (*playing*, *football*), and (*Luke*, *he*). The first two pairs share the same type of relationship, while the latter does not. Furthermore, there exist other pairs of words within the same sentence that exhibit yet another type of relationship, such as (*does*, *not*). However, expecting a single self-attention head to generate features that can identify all of these relationships may prove to be an overly ambitious undertaking. To overcome this challenge, we incorporate multiple self-attention heads that operate in parallel on the same input, with the hope that each head can extract distinctive features that are useful for a specific task. Having gained an understanding of the mathematical underpinnings and effectiveness of multi-head attention, let us now examine a visual representation that aids in concluding our explanation Figure 7.1. The given input is replicated thrice, and subsequently multiplied by distinct sets of weights to generate the query, key, and value. The query and key matrices are multiplied, and the resulting matrix is scaled by dividing it with the square root of the original dimension of the input (depicted by the "Scale" box). The softmax function is applied to the entries of the resulting matrix. Afterward, the matrix is multiplied by the value matrix. The output is then concatenated with other matrices obtained from self-attention heads that have an identical structure but employ different sets of weights. These matrices are generated from the same input and are then multiplied by a final set of weights. This process generates the output of a multi-head self-attention layer. It is noteworthy that the encoding and feeding of input to the multi-head attention layer is crucial in tasks such as Natural Language Processing (NLP). In such scenarios, the position of the sequence elements holds significant importance. Therefore, while encoding the input, it is essential to consider not only the presence of a word but also its position. Hence, before introducing the Transformer architecture, it is imperative to understand the final main component of it.

5.1.4 Positional Encoding

In transformer architecture, positional encoding constitutes a crucial technique utilized to incorporate the sequence's order or position of input tokens, such as words. Since self-attention underpinning the transformer model does not inherently model input token order, positional encoding is essential for communicating this information to the model. Multiple approaches exist for integrating positional encoding in a transformer. One commonly employed technique is to append a vector to each input token's embedding that denotes the token's position in the input sequence. The positional encoding vector is derived from the token's position and the embedding space dimensions and is appended to the token's embedding prior to propagating through the remainder of the model. The design of positional encoding vectors typically ensures that closely located tokens possess similar encoding vectors, while those that are further apart have distinct encoding vectors. This allows the model to distinguish between closely situated and distantly situated tokens, leveraging this information to compute the relationships between input tokens. This technique is fundamental to NLP tasks such as text summarization and machine translation that rely on input token order for interpreting the input text's meaning.

5.1.5 Transformer

Now that we have introduced all of the primary components, we may now discuss the Transformer architecture. The Transformer offers a key advantage in that it can process input sequences of variable length, making it particularly relevant for tasks like machine translation where the length of input and output sequences can significantly differ. Furthermore, the use of self-attention permits the model to capture long-range dependencies in the input, thus playing a critical role in understanding the meaning of the input text. The transformer architecture has thus demonstrated significant effectiveness for natural language processing tasks and has been employed in several state-of-the-art models, such as machine translation and text summarization. The structure of the transformer is characterized by several components, primarily the encoder and the decoder. Following positional encoding, the encoder involves a multi-head self-attention layer where the query, key, and value originate from the input. Subsequently, the position-wise feed-forward neural network applies a multi-layer perceptron to transform all inputs, followed by normalization. A residual connection is added around each of the sublayers of the encoder [29]. The decoder follows a comparable pattern, however, in this case, the input is the target sequence, which is fed into the initial multi-head self-attention layer. In the second multi-head self-attention layer, the output of the encoder serves as input. The normalization layer and the pointwise feed-forward neural network combine the output of the two layers, as shown in the diagram [29].

5.2 FEEL-IT

In this section, I will present one of the models employed in this study. The two models utilized are FEEL-IT and the small version of Flan-T-5. The first was used for lie detection, while the latter model was utilized for answer retrieval and will be introduced separately in a later section. FEEL-IT is a fine-tuned version of a pre-existing model. To provide a comprehensive description, the original model will be briefly outlined, followed by an overview of the fine-tuned version.

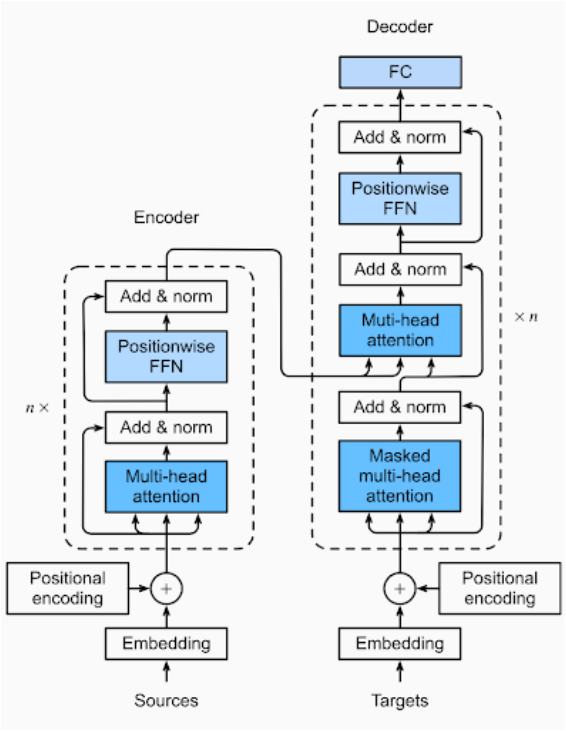

Figure 5.2: Transformer block.

5.2.1 Bidirectional Transformers for Language Understanding (BERT) Model

BERT, an acronym for Bidirectional Encoder Representations from Transformers, is a deep learning model based on the transformer architecture [31]. BERT is composed of a series of transformer blocks arranged consecutively, and it was released in two versions: BERT base and BERT large. BERT base has 12 blocks composed only of the encoder part of the transformer architecture, with 768 hidden sizes, 12 self-attention heads per block, and a feed-forward network composed of 3072 neurons. In contrast, BERT large has 24 blocks with the same structure as the base model, 1024 hidden sizes, and 16 self-attention heads per block, with a feed-forward network of composed 4096 neurons [31]. The term "bidirectional" in the name emphasizes the fact that the transformer block processes the entire input in parallel while considering dependencies from both left-to-right and right-to-left directions in the sequence, similar to a bidirectional RNN. The primary motivation for developing the BERT model was to have a pre-trained natural language processing model that could be fine-tuned on specific tasks. Consequently, BERT was trained using a Masked Language Model pre-training objective, where a random subset of tokens in the input is masked, and the model attempts to predict the original token based on the surrounding context. The goal is achieved by optimizing the cross-entropy loss function between the predicted sequences, which contain the masked tokens, and the original sequences [31]. Additionally, BERT was trained on the "next sentence prediction" task, where input sentence pairs are associated with a label that indicates if one sentence follows the other or not. These two tasks were deemed optimal for building a model that could be useful for various NLP tasks because they require a comprehensive understanding of language and the relationships between words. Further details on the training of the BERT model are available in the original paper.

5.2.2 RoBERTa Model

RoBERTa is an advanced language model developed by the Facebook AI Research team, which distinguishes itself from its predecessor BERT in various ways [32]. One notable distinction is the scale of the training corpus, as RoBERTa is trained on a substantially larger corpus of text data, which enables the model to acquire more robust and generalizable language representations. Furthermore, RoBERTa employs dynamic masking, which involves selecting a distinct set of words to mask in each training iteration. This approach assists the model in making more precise predictions and refining its performance on downstream tasks. Additionally, RoBERTa employs byte-pair encoding (BPE) for word representation, which is a popular NLP data compression technique used to represent words as a series of subword units. This enables the model to handle out-of-vocabulary words more effectively and develop more nuanced word representations. Apart from these technical advances, the RoBERTa model employs more optimized hyperparameters than BERT, including a larger batch size, longer training duration, and higher learning rate. These adjustments facilitate RoBERTa's ability to achieve significant improvements in natural language processing downstream tasks. Also, in this case, the reader is encouraged to go through the paper for more details.

5.2.3 UmBERTo: an Italian Language Model trained with Whole Word Masking

Umberto, an advanced language model trained on large Italian corpora, expands upon the RoBERTa model through the integration of two innovative techniques [33]: SentencePiece and Whole Word Masking. The first technique involves the use of a language-independent subword tokenizer and de-tokenizer, specifically designed for neural-based text processing, to create subword units based on the size of the selected vocabulary and language of the corpus [34]. The latter technique, Whole Word Masking (WWM), applies a mask to an entire word only if at least one of all tokens created by the SentencePiece Tokenizer was originally chosen as the mask. As a result, only complete words are masked, rather than subwords. The integration of these two techniques contributes to Umberto's improved performance in natural language processing tasks.

5.2.4 FEEL-IT: Emotion and Sentiment Classification for the Italian Language

This model is a fine-tuned version of the UmBERTo model for the purpose of distinguishing between joy, fear, anger, and sadness [35]. This particular model was selected for the lie detection task because it has been trained on an Italian language classification task, and its ability to distinguish between four emotions, closely related to the mental conditions which are the topics of the questionnaires analyzed here, is particularly relevant to lie detection. It is sensible to assume that this model has acquired salient features to extract additional contextual meaning from the text, beyond merely detecting the positive or negative sentiment of the sentences.

5.3 Model for answer reconstruction

As previously stated, the task of retrieving a truthful response from a fabricated one can be cast as a regression problem. However, since the problem has been formulated as an NLP task of reconstructing a sequence of sentences given another sequence, it can be considered a text-to-text task. Various text-to-text tasks exist, and multiple text-to-text models have been developed. The most prevalent text-to-text tasks include

- Summarization, in which a text must be summarized while preserving the primary message of the original text;
- Question answering, where the model generates the correct answer to a given question, which can be based on a context or generated from the model;
- Machine translation, where the model must translate a text from one language to another;
- text generation, in which the model must continue an input text with a sequence that is likely to be a continuation of the input, such as continuing "Once upon a time" with "there was a king who had a beautiful and very intelligent daughter";
- Conversational AI, in which the model is capable of conversing with a user.

Ascertaining the truthfulness of a statement by retrieving the honest response, given a deceptive one, poses a unique challenge that does not share many common features with the traditional NLP problems mentioned above. Therefore, the selection of a pre-trained model to fine-tune for this task required consideration of a model with general text-to-text abilities that could be adapted to the reconstruction problem. To illustrate the reasoning behind this selection, let us consider the sentence, "I am afraid that I will never be loved by anyone." If this sentence is a lie, the honest sentence would be, "I am not afraid that nobody will ever love me." The objective of the model is to generate the second sentence based on the first while preserving most of the subjects and the structure of the original sentence. Machine translation was deemed more suitable for this task than question answering since it generates an output in another language that retains the meaning of the input and does not require a question to be asked. The pre-trained model of choice for this task is one that is typically used for machine translation, with pre-training performed on general text-to-text generation. This model possesses the ability to comprehend the meaning of the input, generate an arbitrary length of the output, and naturally maintain the meaning of the input, making it amenable to fine-tuning for the purpose of "translating" dishonest sentences into truthful ones. During fine-tuning, the model learns to preserve most of the sentences in the same language and modify only the way the subjects express themselves concerning a specific topic, acquiring the rules that link dishonest answers to genuine ones.

5.3.1 T-5

The "Text-To-Text-Transfer-Tranformer" model, commonly referred to as T-5, is a type of Transformer architecture that has been developed with the aim of serving as a general-purpose text-to-text model. The model was introduced in [36]. The T-5 model is capable of performing a wide range of tasks by fine-tuning on the specific task at hand, without the need for task-specific architecture modifications or pre-training. This is owing to the fact that the model has been trained on a large dataset comprising diverse text-based tasks. The core of this model is that it is trained on multiple tasks, each of which is transformed into a text-to-text task. The authors of the paper initially introduced a baseline model, which is later improved upon through the exploration of various variants aimed at identifying improvement strategies. The best practices identified through this process are then utilized to train five models that differ solely in size but were trained using the same approach and dataset. The baseline model, which forms the starting point of the model development process, is pre-trained on the "Colossal Clean Common Crawl" dataset, also known as C-4. This dataset is obtained from Common Crawl, an open web archive that provides "web-extracted text" by eliminating non-textual content such as markup from the scraped HTML files. The size of the dataset produced through this process is approximately 20TB per month. To ensure that the database was clean and useful for the purpose, many heuristics were implemented in order to produce the C-4:

- They only retained lines that ended in a terminal punctuation mark (i.e. a period, exclamation mark, question mark, or end quotation mark);
- They discarded any page with fewer than 5 sentences and only retained lines that contained at least 3 words;
- They removed any page that contained any word on the "List of Dirty, Naughty, Obscene or Otherwise Bad Words";
- Many of the scraped pages contained warnings stating that Javascript should be enabled so they removed any line with the word Javascript;

- Some pages had placeholder "lorem ipsum" text; they removed any page where the phrase "lorem ipsum" appeared;
- Some pages inadvertently contained code. Since the curly bracket "{" appears in many programming languages (such as Javascript, widely used on the web) but not in the natural text, they removed any pages that contained a curly bracket;
- To deduplicate the data set, they discarded all but one of any three-sentence span occurring more than once in the data set;
- Finally, "langdetect" has been used to filter only the text with a 99% probability of being English.

The architecture of the baseline model is described below. Regularization is implemented through the use of dropout, with a probability of 0.1. This model is based on the BERTBASE architecture, albeit with twice the number of parameters, as it uses two-layer stacks instead of one. The model's total number of parameters amounts to approximately 220 million. To evaluate the architecture, the authors employed a pre-training strategy, utilizing the C-4. The pre-training involved a specific unsupervised objective, whereby 15% of the tokens in each sequence was dropped. Whenever consecutive tokens were dropped, they formed a dropped span, and each span was assigned a unique sentinel token. The model was then trained, using teacher-forced maximum-likelihood [37], to predict the sentinel token associated with the dropped span in each input sequence. Subsequently, the pre-trained model was fine-tuned on various tasks, all of which were transformed into text-to-text problems to maintain the model's text-to-text nature. During pre-training, the model was trained for $524,288$ steps on the C-4 dataset, employing a batch size of 128 sequences with a maximum sequence length of 512. Where possible, sequences were "packed" into each batch to contain approximately $65,536$ tokens. Pre-training involved approximately 34 billion tokens, a relatively small number compared to other models like BERT and RoBERTa [32]. AdaFactor was used for optimization. The learning rate was set to 0.01 for the first 104 steps and then decayed exponentially using an inverse square root schedule until pre-training was complete. For fine-tuning, the model was trained for $262,144$ steps on various tasks, employing a constant learning rate of 0.001, and a checkpoint was saved every $5,000$ steps. The authors have explored many alternatives to the procedure introduced so far, shortly, the variants of this baseline that have been tested involve:

- changing in the dimension of the batch and/or epoch of training;
- trying variants of unsupervised objective;
- changes in the architecture of the model;
- trying to remove the filtering in the C-4;
- instead of unsupervised pre-training doing multi-task learning with different methods.

In the final model, multi-task learning was employed, which warrants a brief overview. Three distinct multi-task strategies were tested:

- Pre-training the model on multiple tasks in equal proportions;
- Pre-training the model on several tasks with the proportion based on the relative size of each dataset compared to the total dataset;
- Employing a temperature parameter that governed the probability of sampling from each dataset, with the probability increasing with temperature, and eventually becoming equal for all datasets.

However, the models pre-trained using multi-task learning exhibited worse performances than the model pre-trained in an unsupervised manner and fine-tuned on each task. To address this performance gap, the researchers proposed three solutions for the model pre-trained with multi-task learning:

- The first solution entailed fine-tuning the model on the downstream task after pre-training, akin to the unsupervised trained model;
- The second approach involved omitting one task from the multi-task training, pre-training the model on the remaining tasks, and subsequently fine-tuning the model on the omitted task;
- The last strategy entailed omitting the unsupervised objective from the multi-task training.

Among these approaches, the first solution was deemed to be the best [36]. The results of those experiments made the researchers come out with this strategy to develop the final model:

- By employing a corrupted span objective, which involves corrupting spans of 15% of each sequence, the average number of tokens per span was found to be 3;
- The models are subjected to pre-training for a duration of one million steps while utilizing a batch size of 2048 sequences with a length of 512. This process generates a total of approximately 1 trillion pre-training tokens, which is around 32 times higher than that of the baseline model;
- Pre-training the model using a multi-task learning approach that leverages proportional probability sampling for each task in addition to unsupervised learning. While both methods yield comparable outcomes, the former allows for the evaluation of the model's downstream task performance beforehand;
- Reduction on batch size for GLUE and SuperGLUE tasks while fine-tuning;
- The baseline model was evaluated using greedy decoding, which means that each token was predicted according to its unconditional probability, while for tasks with long outputs, beam search improved the performance, so the final models were used with a beam width of 4 and length penalty of 0.6 [36].

Five versions of the architecture were trained using the strategy above:

- **Base.** The desing of the encoder and the decoder has been made to make it similar to BERTBASe, with 12 blocks, (each block comprising self-attention, optional encoder-decoder attention, and a feed-forward network). The feed-forward networks in each block consist of a dense layer with an output dimensionality of $d_{ff} = 3072$ followed by a ReLU nonlinearity and another dense layer. The "key" and "value" matrices of all attention mechanisms have an inner dimensionality of $d_{kv} = 64$ and all attention mechanisms have 12 heads. All other sub-layers and embeddings have a dimensionality of $d_{model} = 768$.[36] It has approximately 220 million parameters.
- **Small.** The researchers considered a smaller model that scales down the baseline by utilizing $dmodel = 512$, $d_{ff} = 2,048$, 8-headed attention, and only 6 layers for both the encoder and decoder. This variant has around 60 million parameters.
- **Large.** As the baseline employs a BERTBASE-sized encoder and decoder, another variant is considered, which uses an encoder and decoder similar in size and structure to BERTLARGE. This variant employs $d_{model} = 1,024$, $d_{ff} = 4,096$, $d_{kv} = 64$, 16-headed attention, and 24 layers for both the encoder and decoder, resulting in roughly 770 million parameters.

- **3B and 11B.** Two additional variants are considered to explore the potential performance of larger models. In both cases, $d_{model} = 1024$, a 24-layer encoder and decoder, and $d_{kv} = 128$ are utilized. For the "3B" variant, $d_{ff} = 16,384$ with 32-headed attention is used, resulting in a model with around 2.8 billion parameters. For "11B," $d_{ff} = 65,536$ with 128-headed attention is used, resulting in a model with approximately 11 billion parameters. The choice to increase d_{ff} specifically is because modern accelerators, such as the TPUs employed in training the models, are most efficient for large dense matrix multiplications such as those in the Transformer's feed-forward networks.

In this work, I used a model that is the same architecture as the small version of the T-5, namely the Flan-T5 small. In the next section, we shall see how the latter differs from the first.

5.3.2 Flan-T5

Flan-T-5 has been demonstrated to surpass T-5 in every aspect [38]. With the same number of parameters, it has been trained on over 1,000 tasks in various languages, including Italian. Therefore, the small version of Flan-T-5 was ultimately chosen. The primary improvements of Flan-T-5 are attributed to instruction fine-tuning, which involves training the model to generate text in response to a text that expresses an instruction. For instance:

1. **Input**: "Answer to the following question. Who was the first President of America?
2. **Output**: George Washington

In this case, the instruction "Answer to the following question" is provided as input, followed by the actual question that needs to be answered. Another crucial component of instruction fine-tuning is chain-of-thought fine-tuning, which requires the model to provide the answer to a question preceded by a step-by-step explanation of the reasoning that led to that answer. For example:

- **Input** "Answer the following question by reasoning step-by-step. The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?"
- **Output** "The cafeteria originally had 23 apples. They used 20 apples for lunch, leaving them with 3 apples. They subsequently bought 6 more, resulting in a total of 9 apples."

The paper demonstrates that this type of fine-tuning significantly enhances the models' performance, particularly in the few-shot and zero-shot learning frameworks [38].

1. **Input**: "Answer to the following question. Who was the first President of America?
2. **Output**: George Washington

In this case, the instruction "Answer to the following question" is provided as input, followed by the actual question that needs to be answered. Another crucial component of instruction fine-tuning is chain-of-thought fine-tuning, which requires the model to provide the answer to a question preceded by a step-by-step explanation of the reasoning that led to that answer. For example:

- **Input** "Answer the following question by reasoning step-by-step. The cafeteria had 23 apples. If they used 20 for lunch and bought 6 more, how many apples do they have?"
- **Output** "The cafeteria originally had 23 apples. They used 20 apples for lunch, leaving them with 3 apples. They subsequently bought 6 more, resulting in a total of 9 apples."

The paper demonstrates that this type of fine-tuning significantly enhances the models' performance, particularly in the few-shot and zero-shot learning frameworks [38].

6

Transfer Learning

In the present discussion, we aim to provide a comprehensive understanding of Transfer Learning and justify its use in Natural Language Processing (NLP) models. Transfer learning is a widely utilized technique in various domains such as computer vision, speech recognition, and natural language processing, wherein a pre-trained model is adapted to perform a related task. This approach has been recognized for its ability to enhance performance and decrease the amount of labeled data required for training, thus gaining popularity among researchers ([39]; [40]; [41]).

There exist several methods of transfer learning, including fine-tuning, feature extraction, and multi-task learning [42]. Fine-tuning entails updating the pre-trained model's weights on a new task using a minimal amount of labeled data [43]. Feature extraction involves utilizing the pre-trained model's learned features as input to a new model trained on a new task [44]. Multi-task learning, on the other hand, trains a single model to perform multiple related tasks concurrently [45].

Transfer learning allows researchers to leverage the knowledge acquired from extensive datasets and pre-trained models to enhance the performance of a new task with limited labeled data [46]. This is particularly useful in cases where collecting and labeling a large dataset is infeasible or cost-prohibitive. Transfer learning has been employed in a diverse range of applications, such as image classification [47], object detection [48], and machine translation [49]. Similarly, pre-trained language models have been utilized to improve the performance of machine translation systems [50] . Nonetheless, the effectiveness of transfer learning is influenced by several factors such as the relatedness of the source and target tasks [41], the quantity and quality of labeled data available for the target task [47], and the choice of pre-trained model and transfer learning approach [40]. Researchers have also proposed diverse techniques for selecting and adapting pre-trained models for transfer learning based on task similarity, feature representation, and network architecture [44]; [43]).

Despite its success, transfer learning has limitations and challenges that hinder its application. One of the primary limitations is the pre-trained model's unsuitability for the target task, particularly if the tasks are significantly different, or the pre-trained model has not been trained on a diverse enough dataset [42]. Moreover, transfer learn-

ing's performance may decline as the gap between the source and target tasks increases ([51]).

6.1 Fine-Tuning

Fine-tuning has emerged as a prominent method for transfer learning in the field of natural language processing (NLP), whereby a pre-trained model is repurposed or adapted for a related task using a limited quantity of labeled data (Howard and Ruder, 2018). When training a model from scratch, the parameters are initialized with random values, followed by the application of the backpropagation algorithm to update those values, thereby rendering them useful for the completion of the task at hand. Fine-tuning a pre-trained model essentially follows the same process, with the exception that the parameters are not initialized at random but with the values they previously attained at the end of a training session for another task. This methodology is beneficial when the parameters acquired during the initial task are similarly applicable to the secondary task, thereby streamlining the iterative process and accelerating the development of a high-performing model for the secondary task.

The advantages of this approach are multiple, under the assumptions that: The suitability of fine-tuning a pre-trained model for a related task is contingent upon two factors:

- First, the tasks must exhibit sufficient similarity to enable the utilization of the original parameter configuration for the new task without excessive modification;
- Second, the model must possess sufficient capacity to learn the new task, such that the level of complexity of the task is not greater than that of the original task for which the model was trained, precluding the need for any additional ensemble techniques.

These considerations are essential for ensuring the effectiveness of fine-tuning in natural language processing applications. As previously noted, one of the primary benefits of fine-tuning is its potential to reduce computational costs, as it necessitates fewer iterations. Furthermore, a pre-trained model may be capable of achieving comparable results to a model trained from scratch on the same task, but with substantially fewer data, which can result in significant computational savings. This is particularly important when data availability for the fine-tuning task is limited, or additional data collection is not feasible. However, the primary rationale for employing fine-tuning is typically a scarcity of resources necessary to train a high-performing model from scratch, and the availability of a pre-trained model suitable for the task at hand. If a pre-trained model is able to save time during training, and incurs comparable computational costs during inference when compared to the best alternative, it should be favored over a model trained from scratch, provided that it does not perform worse even after fine-tuning.

6.1.1 Why fine-tuning is useful in this case

The rationale for transforming the original data into textual format pertains to a desire to evaluate the performance of state-of-the-art natural language processing (NLP) models in the context of lie detection and answer reconstruction. The decision was primarily driven by the need to leverage the models' ability to comprehend the meaning of textual data, which is a prerequisite for identifying fake responses and attempting to recover authentic ones. Additionally, Transformer-based architectures are specifically engineered to capture the interdependence among

sequence elements, which constitutes the fundamental basis for understanding sentence semantics. In light of this, such architectures could potentially assist in our task, as the response can reasonably be viewed as a sequence.

6.1.2 From numbers to texts, and then to (different) numbers, why?

A model trained using the framework proposed in this study can effectively handle all questionnaires, eliminating the need for multiple models to handle different questionnaires. This is possible because the model relies solely on the meaning of the text and not on the numerical values assigned to the answers, which may vary across questionnaires. This is an often overlooked aspect of the model but is, in fact, its most significant advantage. It allows for future research into training a model to detect deception in the generic text produced by individuals and a model that reconstructs truthful responses from the same text. Tokenization, using SentencePiece in this case, is performed to transform the text into numbers. The tokenization process ensures that each token, considered atomic, is transformed into a specific number that uniquely identifies it. Additionally, Transformer-based architectures have a positional encoding layer that further transforms the sequence, allowing the model to distinguish between the same words in different positions in the sequence. Encoding each possible answer to each question of every questionnaire could achieve comparable results without transforming the responses into text. However, this approach is not pursued in this study due to its limitations. It is not scalable to unseen questionnaires/questions, it does not allow for fine-tuning, and it requires significantly more data than what is available for this study.

7 Results

After having established the validity of the proposed approach, the subsequent sections will present the experimental setups and results of both Lie Detection and Answer Reconstruction. Each of these setups and results will be described in detail, to provide a clear and comprehensive explanation that will enable the reader to comprehend the achievements and limitations of this thesis. These discussions will lay the groundwork for the conclusions that will be drawn in the subsequent sections. To implement the experiments of this work I used the hugginFace python libraries for NLP tasks. All the code was executed on the free version of Colab using the GPU runtime for training and generating. In both Lie Detection and Answer Reconstruction, I performed some preliminary experiments with different train-test splits, and the hyperparameters that revealed superior performance were chosen to perform the experiment presented here. The train-test split used for the result presented here was chosen randomly.

7.1 Lie detection

7.1.1 Setup

For this experiment, 90% of the dataset was utilized for training, while the remaining 10% was reserved as the test set. The FEEL-IT model was fine-tuned for three epochs using a Polynomial Decay strategy with an initial learning rate of $5e-5$, terminating at a learning rate of 0.0, and the number of steps was determined by multiplying the number of epochs by the number of batches. The selection of three epochs was substantiated by preliminary experiments, which demonstrated that this value was sufficient to achieve maximum accuracy without overfitting. The utility of Polynomial Decay stems from its systematic and effective approach to modifying the learning rate during training [52]. By decreasing the learning rate over time, the model can converge more seamlessly and prevent overshooting of optimal weights. Moreover, Polynomial Decay can mitigate the risk of model entrapment

Model	Accuracy
FEEL-IT	95.02%

Table 7.1: Test accuracy.

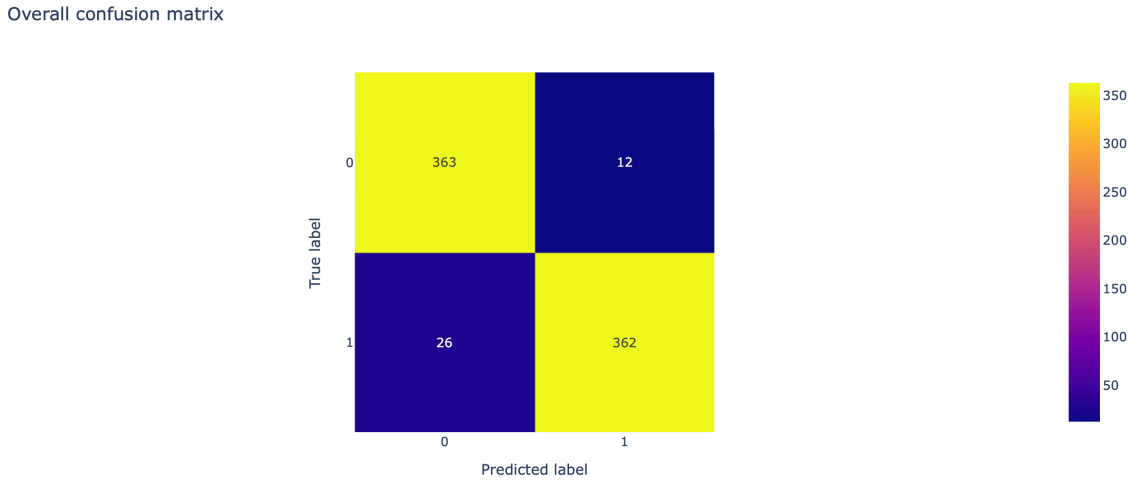

Figure 7.1: Overall confusion matrix.

in local minima or plateaus during training and enhance the generalization performance of the model. However, selecting the appropriate polynomial function and hyperparameters for a given task is still an active research field. Therefore, in this study, the default TensorFlow implementation was employed without any further investigation. The batch size was set to 8 to ensure computational feasibility with the available resources even though has been established that larger batch sizes are advantageous for Large Language Models.

7.1.2 Results

The results are shown below for the test set Table 7.1 Given the encouraging results obtained, it is deemed necessary to conduct a more detailed analysis of the model's performance. The investigation shall begin with an examination of the overall confusion matrix presented in Figure 7.1. The matrix reveals that the model is impartial towards either class, with the honest class represented by 0 and the dishonest class represented by 1. Another noteworthy aspect to highlight is the varying accuracy of the model w.r.t. different types of questionnaires. As shown in Figure 7.2, the performance of the model is not uniformly distributed across all types of questionnaires, despite being generally high for all of them. The reason for this discrepancy is not entirely clear, but it appears that the model performs better on questions with a larger number of available samples. However, the IESR questionnaire is an exception to this behavior. In order to identify the source of the 5% error rate, further investigation of the model's performance is warranted. To this end, we propose to partition the test set into two groups based on whether the associated scores fall below or above the established cut-off threshold. We remind the reader that the cut-off is the value at which responses are deemed indicative of the mental state being assessed by a given

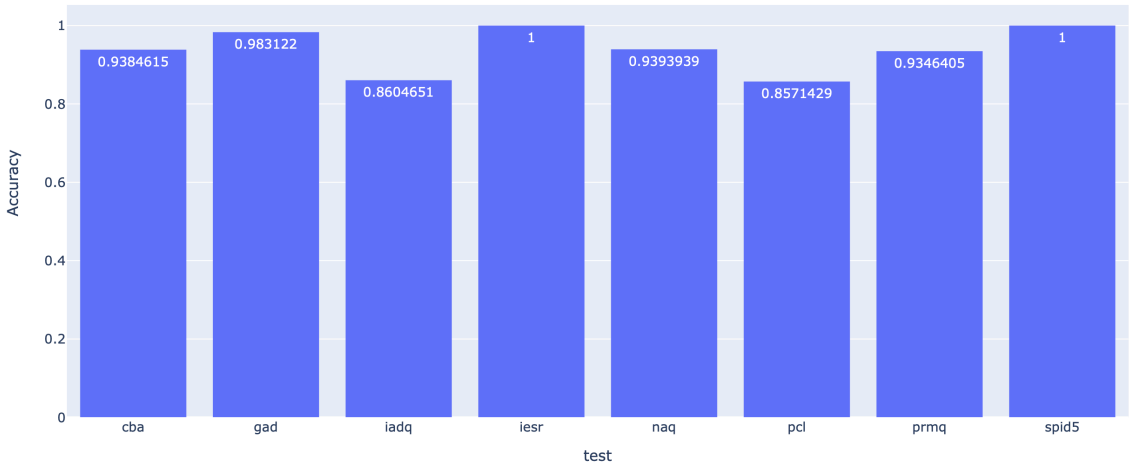

Figure 7.2: Accuracy type-wise.

questionnaire. By examining the confusion matrix for the samples falling below the cut-off threshold, as depicted in Figure 7.3 we can see where the model falls short. The issue at hand pertains to the model's ability to distinguish certain dishonest answers that do not surpass the cut-off score. This is due to the fact that these answers may appear to be honest, as participants were instructed to respond dishonestly to emulate a specific mental state across all experimental datasets. Looking at the group that did score above the cut-off Figure 7.4 we can see that the model is surprisingly able to classify correctly the honest answers, mistaking only one dishonest response. In conclusion, the findings suggest that the model has acquired the ability to classify as dishonest, responses that malignantly amplify psychological symptoms, but its effectiveness decreases when confronted with deceptive answers that lack clear indications of mental disorders.

7.1.3 Performance on unseen questionnaires

So far, the performance of the model has been evaluated utilizing a test set where the responses are given to the same questionnaires to which also the responses of the training set were given. This allowed us to understand what the model is capable to learn from the dataset, but it does not give any clue on how the model would perform on unseen types of questionnaires. To gain some insight into how the model would perform in this last case, I performed 8 experiments where in turn each type of questionnaire is selected and all the responses given to that questionnaire are used as a test set, and all the others constitute the training set. In each experiment, a new instance of the FEEL-IT model is fine-tuned for 1 epoch with the same strategy as the first experiment above discussed. The results are summarized in Figure 7.5. We can see that the performances generally (with the exception of NAQ) decreased by a few percentage points but remained satisfying, except for PCL and PRMQ where the drop was more accentuated, and IADQ where the performance decreased to chance level. A drop of a few percentage points is comprehensible because each questionnaire has its peculiarity that the model could not learn when it is not fine-tuned on it. Examining the case of PCL, we can see that the decrease in percentage points is slightly higher than the more fortunate cases discussed just earlier, and not comparable with the significant reduction that occurred

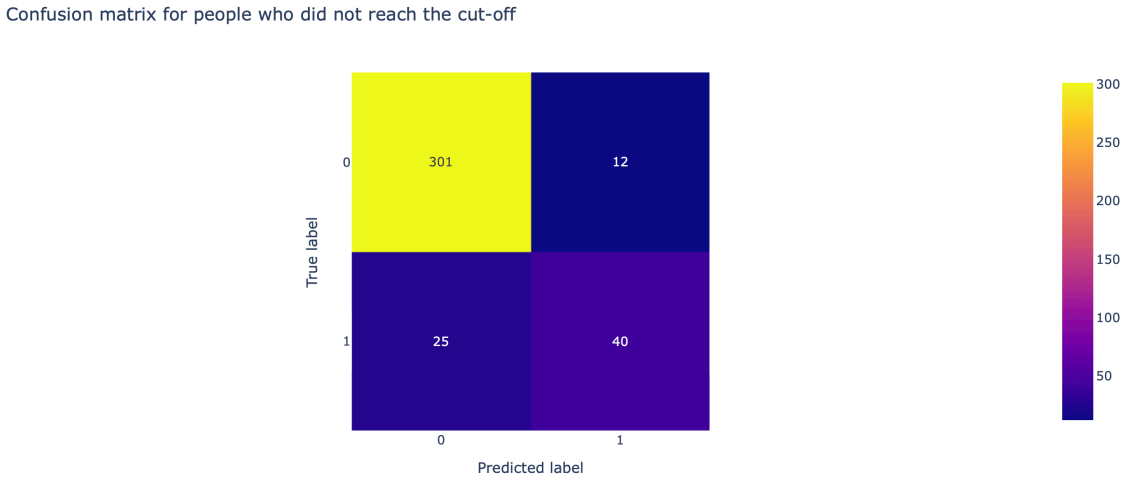

Figure 7.3: Confusion matrix for whom did not score above the cut-off.

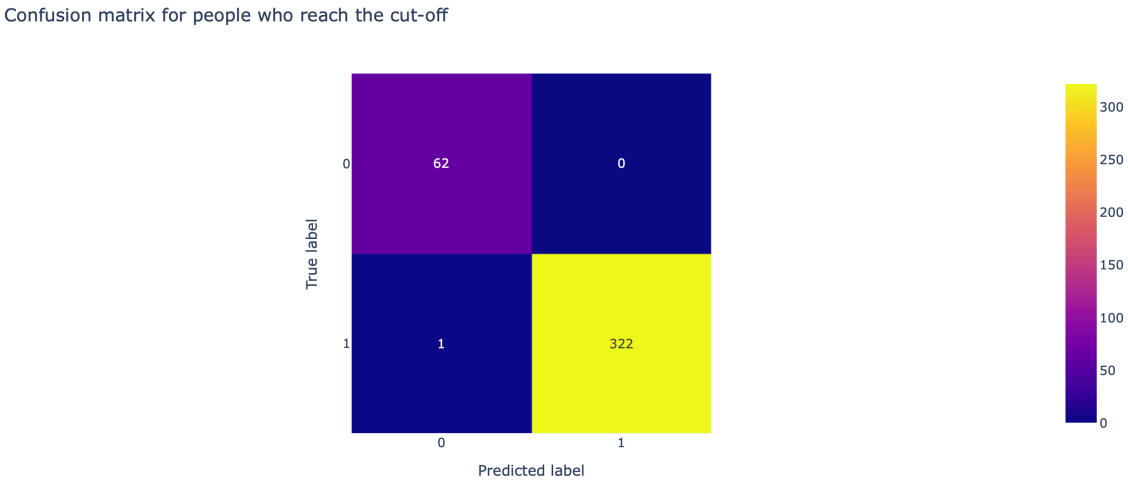

Figure 7.4: Confusion matrix for whom scored above the cut-off.

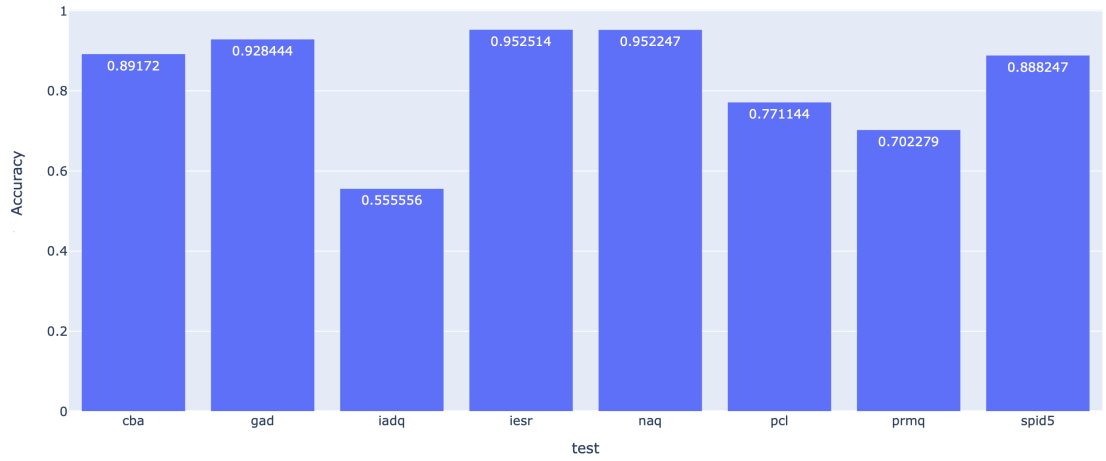

Figure 7.5: Train and Validation loss

for PRMQ and IADQ. One possible explanation of this phenom is that the responses related to PCL are hard to classify in general even though they present similar features to the other responses. So the performance decreases a little more than the others. This hypothesis is supported by the fact that in the case responses to PCL are included in the training set, the model has its worst performance when tested on unseen responses to the same questionnaire. This could be mainly due to a combination of both the intrinsic difficulty of those samples to be classified and the scarcity of those samples compared to the portion of presence of answers to the other questionnaires. In the case of PRMQ, the nature of this questionnaire is different from all the others, because while the last mentioned try to assess mental disorders that are all associated with common symptoms like depression and anxiety, and/or common causes like being the victim of some kind of traumatic bad event, PRMQ tries to assess the memory capacity of the subject who is to respond. In spite of the instruction given to the subjects of that questionnaire being the same as all the others, i.e. to exaggerate the answers to simulate the presence of the mental problem, the semantic meaning of this exaggeration deviates from the topics treated by all the other questionnaires. This possibly explains why the performance of the model that has been trained also on responses to PRMQ is much higher compared to the one that has not. In the first case, the model learned the mechanism of exaggeration also in the context of memory functioning, in other words, it has learned that exaggeration can occur in different semantic contexts, hence it became easy for it to classify correctly those responses. In the second case, the model still learned to recognize the exaggeration mechanism but only in one semantic context. As a consequence, it gets deceived more easily when the topic treated is too far from the one it has been trained on. The IADQ questionnaire is the second most difficult to classify when the model is trained on it, and it brings the model to chance level when this is not trained on it. The reason why this questionnaire is so difficult to treat in both cases is because of its shortness which makes it difficult to classify it. It is a well-known fact in psychology that the larger the number of items of a questionnaire the better it asses the condition of the subject and the easier is to tell if the response is honest or not, and vice-versa [53]. So, the exaggeration mechanism that occurs in a short questionnaire is much different from the exaggeration mechanism that occurs in a longer questionnaire Therefore the model necessarily needs to be trained on the IADQ in order to learn the ability to classify samples coming from that dataset because it is a more difficult task than classifying responses coming from longer questionnaires.

7.2 Answer reconstruction

7.2.1 Setup

In the Answer Reconstruction task, the dataset was divided into training and testing sets with the same proportion as that used for the Lie Detection task. The optimization algorithm adopted was the AdaFactor with weight decay, which employs an initial learning rate of $5e-4$. This value was found to be superior to the initial learning rate of $5e-5$ in preliminary experiments and has been observed to lead to a significant reduction in loss. A weight decay coefficient of 0.01 was employed in the optimization process. The loss function utilized in this task is the same as that used during pre-training of the model[38]. The model was fine-tuned for 10 epochs, but as we can see from Figure 7.6 two epochs are enough to achieve the most significant loss reduction. To evaluate the model, two strategies were employed to generate the responses: Greedy Search and Diverse Beam Search, to asses if the generation strategy could affect the results.

- Greedy strategy: in this strategy, each word of the sequence was chosen as it was the most probable, therefore this strategy is called greedy because it maximizes the probability of the single current word, without considering the sequence as a whole. This strategy could potentially lead to sequences that do not make sense or are somehow imprecise, especially when generating long sequences;
- The other strategy is called Diverse Beam Search which is an improvement to the classic Beam Search. Beam Search for each position of the sequence selects the n most probable words, thus it expands all the paths so created and then outputs the most probable sequence. This strategy leads many times to almost identical sequences, resulting in a waste of time and computational resources. Diverse Beam Search enforces diversity among all the alternative sequences generated by means of a diversity-augmented objective [54]. This strategy divides all the alternative sequences into m groups and enforces maximum diversity among sequences from different groups. In this experiment $n = m = 5$

. Despite their diversity, the two strategies gave place to the same responses. This sustains all the explanations and the conclusion made later in this work.

7.2.2 Results

In order to provide a comprehensive evaluation of the Answer Reconstruction model, relying solely on the loss function is insufficient, as it does not provide a sense of the quality of the generated responses. Therefore, to assess the model's performance, for each sample in the test set, the reconstructed answer generated was converted back to the original scale of its questionnaire, and the accuracy, Root Mean Squared Error (RMSE), and Mean Absolute Error (MAE) were calculated w.r.t. the corresponding honest answer. The resulting metrics were then used to evaluate the quality of the reconstruction. The summary statistics of these metrics are presented in Table 7.2, which provides an overview of the model's performance and aids in its interpretation.

The reader should keep in mind that all the considerations made to interpret the results in the case of Lie Detection are most likely to hold also in the case of Answer Reconstruction. The accuracy of the reconstruction indicates the number of items that were reconstructed correctly, and it is observed that the average accuracy is approximately 45%. This is mainly due to the fact that the model primarily generates reconstructions that correspond to the 2 lowest (or rarely highest) numbers of the corresponding scale, and it prefers the lowest (highest)

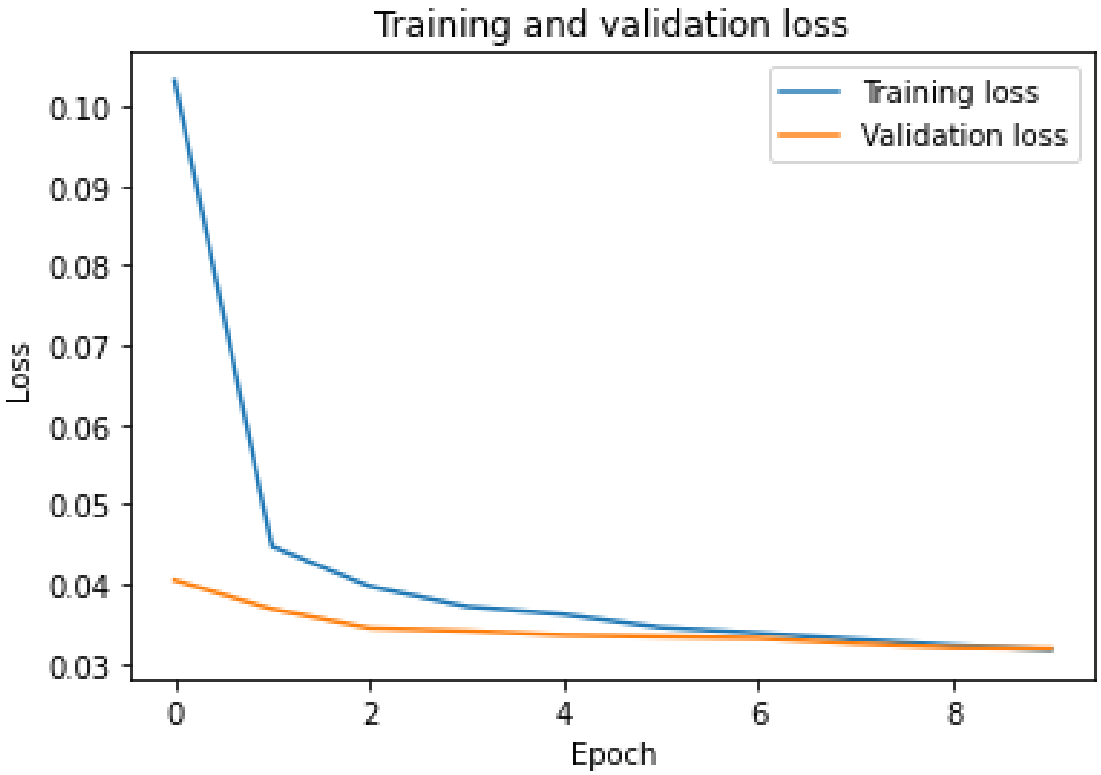

Figure 7.6: Train and Validation loss

Statistic	Accuracy	RMSE	MAE
Mean	44.81%	1.162	0.84
Stand. Dev.	22.6%	0.495	0.471
Max.	100%	2.55	2.375
Min.	0.00%	0.00	0.00
Median	45.00%	1.09	0.733
1 Quartile	26.67%	1.08	0.478
3 Quartile	60.00%	1.533	1.187

Table 7.2: Statistics of the metrics

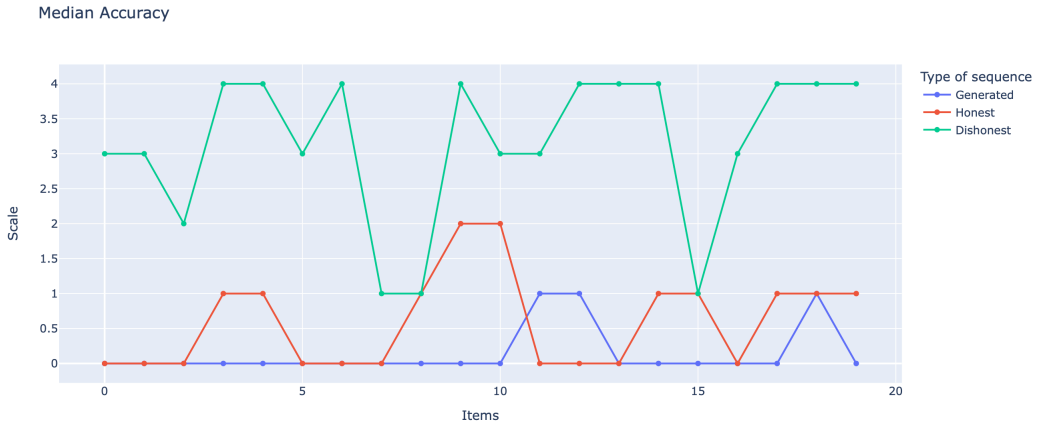

Figure 7.7: Generated vs Honest vs Dishonest response with median accuracy

between the two. This implies that it mostly fails to capture the nuances of each honest answer. However, this is not necessarily a major concern since, as previously demonstrated, the honest answers mainly comprise the lowest numbers on the scales. Therefore, the model correctly captures this trend. It should also be noted that when interpreting the results of a questionnaire for a single patient, evaluating the general trend of the answer is more crucial than assessing the response given to every single item, and the model performs well in this regard. Accuracy, in this case, does not reflect how much the model is wrong for every part of the answer the model reconstructs. To evaluate this, MAE, which gives equal weight to all errors, and RSME, which emphasizes larger errors, were employed. In both cases, the average error of the model is around 1 place on the scale, which is not too bad since there is no considerable difference in meaning between two consecutive numbers on the scale. To better understand the performance of the model, it is useful to compare some of the generated responses with the respective honest and dishonest responses, after converting them back to the original scale. What we can see looking at one of the samples with the accuracy equal to the median value Figure 7.7 . We can see how the model prediction is near to the honest answer but fails to capture the details. The findings suggest so far that the model is not always capable to reconstruct precisely anything that deviates itself from the general trend, therefore to further inspect this behavior it is useful to compare the RMSE with the standard deviation of each honest response from their respective average score on the scale as a measure of the presence of peculiarities in the honest response. Figure 7.8. As we can see, the higher the standard deviation of an honest response the higher will be the error that the model will make in reconstructing it. These findings confirm that the model is capable of providing a plausible reconstruction of the overall condition of the patient but falls short in reconstructing the finer details, especially if they deviate much from the average score. Doing the same comparison but with the standard deviation of the dishonest responses, we can see that there is not any type of correlation with the errors made by the model Figure 7.9 One hypothesis on why the model fails to reconstruct more precisely the honest answer is because the way the responders falsifies one aspect of their mental state is not correlated with the mental state itself, but instead it is correlated with the scenario in which they find themself when they are compiling the questionnaire. As result, their dishonest responses are the reflection of what they think is more convenient to appear in a specific situation and it could be the case that this has almost nothing to do with what the answers would be if the responders were to answer hon-

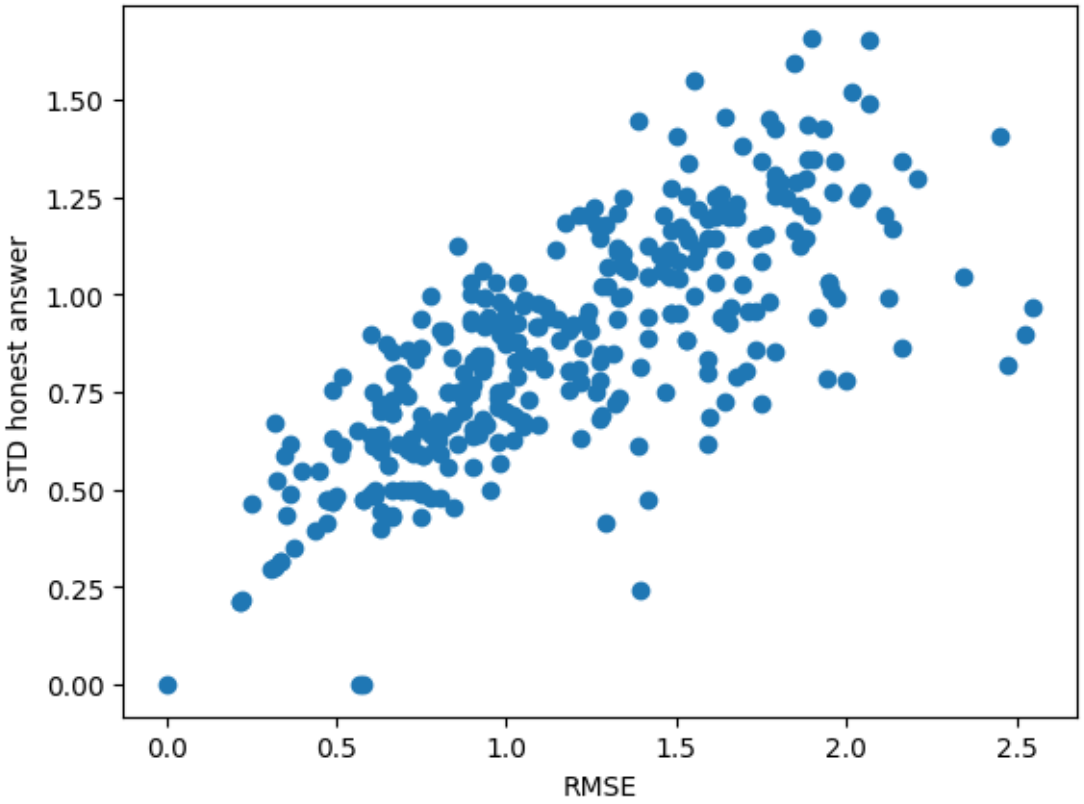

Figure 7.8: RMSE of the generated responses vs Standard deviation of the honest responses

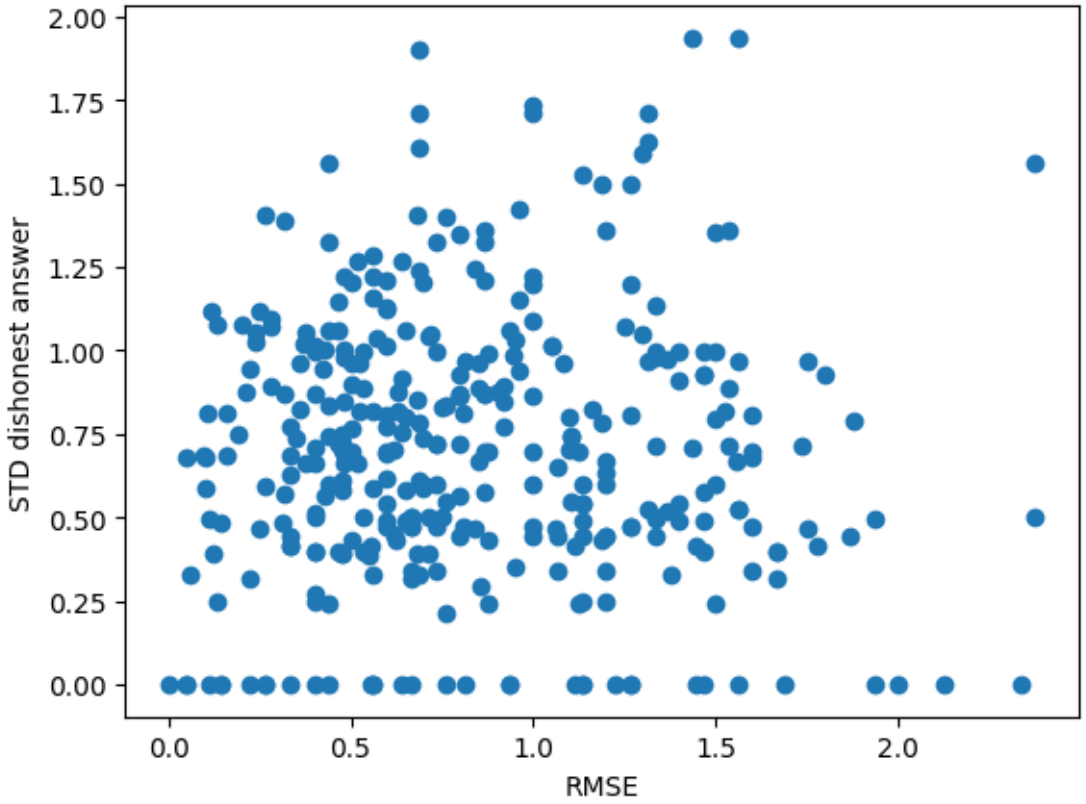

Figure 7.9: RMSE of the generated responses vs Standard deviation of the dishonest responses

estly. In the works where the data comes from, all the subjects were instructed to exaggerate a specific pathological condition, while it is not present at all. Hence, the model successfully captured the phenom of exaggeration, and tries to reduce it in the reconstruction, but it cannot retrieve the details of the honest answers because there is not enough information in the data, that links more accurately the dishonest and honest answers, to be learned by the model. The only link between the two groups of responses is provided by the common scenario where all the pieces of data were extracted. Therefore it is the scenario to rule out the relationship between dishonest and honest responses, and how the subject decides to lie in a specific scenario simply does not depend on what they would say if their interest was to accurately report the truth. In fact, since the scenario motivates the subjects to exaggerate the pathological condition, they will be exaggerating their answers regardless of how they actually feel about the items, and as result, the model is able to tell that in general, the answers have been exaggerated, but it lacks in telling exactly in which measure. One possible way of improving the performance of an LLM trained to reconstruct honest answers could be to include as input personal information about the responder. Furthermore, assuming that the previous hypothesis is true, it would be necessary to train the model on data coming from different scenarios and provide explicitly the context representing the scenario as input to the model. This could help in better leveraging the existing capabilities of the LLM, and at the same time, it would add more features that are most likely to improve the performance, at least in some cases. In general, more data would surely benefit the quality of the model, at cost of more training time. Another strategy to improve the performances would be to use a larger model, as it has been shown that larger models trained on the same dataset perform better on the same tasks [36].

8

Conclusion

To conclude this work, I would like to summarize the results shown and discuss the take-home messages of this work, as well as the limitations and possible future works. To begin with lie detection, in this work I assessed the capabilities of a large language model to be fine-tuned to perform this task. I explored the qualities and the defects of the fine-tuned model. Given enough data, a Large (enough) Language Model can learn all the relevant features relevant to classify correctly most of the samples. Some of the most important features used by the model are implicit in the distribution where the samples come from, and not explicitly encoded in the samples them-self. One prominent feature of this kind is the experimental environment in which all the samples were created. The common instruction in all experiments was to exaggerate the responses, therefore the model has learned this pattern in which exaggerated responses are associated with dishonesty, and overall lower responses are associated with honesty. As we saw, the model is still able to classify correctly honest answers that have a higher score w.r.t. the average honest responses, but it gets deceived by dishonest answers with a relatively low score. This phenom suggests that the exaggeration mechanism that is the consequence of the instructions given during the experiments highly affects the internal representation of the model of honesty and dishonesty. Therefore the model is likely to not be effective when classifying dishonest responses coming from other contexts. The reason why the exaggeration mechanism affected the fine-tuning so drastically is that it is the main factor distinguishing dishonest from honest responses, as we saw also in the exploratory analysis. Consequently, it is mandatory to collect and build the training set for this fine-tuning task with responses coming from many different contexts, where the diverse participants are provided with diverse instruction/ incentives to lie in a specific way, to obtain a better model in the end. Another crucial aspect is the semantic topic treated in the samples, which needs to be as variegate as possible, otherwise will be difficult for the model to classify all those answers that do not share semantic meaning with the training samples, as we saw to be the case with PRMQ. Clearly, this diversity is not achieved in the dataset used in this work, thus the fine-tuned model obtained in this work is far from deployable in all those scenarios where responses come from questionnaires that assess different mental conditions than the ones assessed in this dataset. Nevertheless, in this work the capacity of the LLM to adapt to diverse semantics meaning to perform lie detection

has been proved as expected, after all, a large language model is by definition able to perform, with some kind of performance, NLP tasks on inputs with diverse semantic meaning due to their training on a vast variety of texts. Lastly, one feature that has been revealed as crucial is the length of the sequences: a more long response is easier to classify than a shorter one, in virtue of this, it is important to fine-tune the model on responses with high variability of length to obtain a more robust model. To fully appreciate the informative value of this work of lie detection on psychological questionnaires, yet its limitation and imagine possible future works, one has to recall that here all the responses in the form of text were artificially created starting from the numbers of the scales of each questionnaire. This transformation has been done with the intent of generating a sentence with a semantic meaning as similar as possible to what the subjects of each experiment would have written if they were to answer each item by writing a text that expresses how they feel about that item, rather than just inserting a number on a scale. But obviously, given also the fact that no other information than the responses were given about each respondee, it was impossible to reconstruct a personalized textual response starting from the numerical one. Therefore, the criteria of conversion from numbers to text were all identical for each response to the same questionnaire. This procedure maintains the same level of information contained in the numerical responses, but from an NLP perspective, an LLM model that is fine-tuned on this type of dataset learns something fundamentally wrong: that a person who is expressing about a certain topic honestly, does it the same way as if she/he was being dishonest. To make an example, if the context motivates a person to exaggerate the response of the first item of our Short Random Example, that was "I enjoy public events", this person would probably respond 4 to this item, which corresponds to "I really enjoy public events". If who is responding is actually shy and does not enjoy public events, the honest response would be 1, which corresponds to "I do not enjoy public events". As we can see, the only difference is that in place of the word "really", there is "do not", and then the phrase is identical. As argued before, the way people lie involves different parts of the brain than the parts involved when they say the truth. Furthermore, the way people lie is affected also by the circumstances and cultural conditions they find themself. Last but not least an important role is played by the goal of the liar when she/he is lying, besides her/his personality. All of those things are sure to affect the way people express a concept when they can freely produce a text, arbitrarily long, to do so, giving rise to a diverse text depending if they are being honest or not. All this information is simply thrown away with a numerical answer, and it could be partially reconstructed only taking into account also the answers to the other items of the questionnaire. As it should be clear now, this constitutes the biggest limitation of any LLM fine-tuned on this dataset artificially created. This consideration allows us to appreciate the potential that the approach showed in this work would have if applied to real natural language, as there are many more features to be discovered by LLM to correctly discriminate between truthful and deceiving responses, given the fact that all the improvements suggested in this section should be applied as well. The same considerations hold for the task of Answer Reconstruction, recalling also the suggestion given in the section results. Overall, in this work, LLMs models have been shown to be able to successfully extract the information contained in a textual dataset and use it also for non-conventional NLP tasks like Lie Detection and Answer Reconstruction. This work is meant to be one of the first building blocks to the use of AI applied to those tasks, as it naturally highlights all the next steps to build a successful lie detector based on natural language, and also highlights the usefulness and the limitations, hence the improvement areas of psychological questionnaires.

References

[1] P. Ekman, *Telling lies: Clues to deceit in the marketplace, politics, and marriage (revised edition).* WW Norton & Company, 2009.

[2] G. Ganis and J. P. Keenan, "The cognitive neuroscience of deception," *Social Neuroscience*, vol. 4, no. 6, pp. 465–472, 2009.

[3] A. Vrij, S. Mann, S. Kristen, and R. P. Fisher, "Cues to deception and ability to detect lies as a function of police interview styles," *Law and human behavior*, vol. 31, pp. 499–518, 2007.

[4] B. M. DePaulo, D. A. Kashy, S. E. Kirkendol, M. M. Wyer, and J. A. Epstein, "Lying in everyday life." *Journal of personality and social psychology*, vol. 70, no. 5, p. 979, 1996.

[5] A. K. Gordon and A. G. Miller, "Perspective differences in the construal of lies: Is deception in the eye of the beholder?" *Personality and Social Psychology Bulletin*, vol. 26, no. 1, pp. 46–55, 2000.

[6] L. Saxe, "Lying: Thoughts of an applied social psychologist." *American Psychologist*, vol. 46, no. 4, p. 409, 1991.

[7] C. F. Bond Jr and B. M. DePaulo, "Accuracy of deception judgments," *Personality and social psychology Review*, vol. 10, no. 3, pp. 214–234, 2006.

[8] A. Vrij, R. Fisher, S. Mann, and S. Leal, "Detecting deception by manipulating cognitive load," *Trends in cognitive sciences*, vol. 10, no. 4, pp. 141–142, 2006.

[9] S. E. Christ, D. C. Van Essen, J. M. Watson, L. E. Brubaker, and K. B. McDermott, "The contributions of prefrontal cortex and executive control to deception: evidence from activation likelihood estimate meta-analyses," *Cerebral cortex*, vol. 19, no. 7, pp. 1557–1566, 2009.

[10] K. Suchotzki, B. Verschuere, B. Van Bockstaele, G. Ben-Shakhar, and G. Crombez, "Lying takes time: A meta-analysis on reaction time measures of deception." *Psychological Bulletin*, vol. 143, no. 4, p. 428, 2017.

[11] V. A. Gombos, "The cognition of deception: The role of executive processes in producing lies," *Genetic, social, and general psychology monographs*, vol. 132, no. 3, pp. 197–214, 2006.

[12] D. B. Buller, J. K. Burgoon, A. Buslig, and J. Roiger, "Testing interpersonal deception theory: The language of interpersonal deception," *Communication theory*, vol. 6, no. 3, pp. 268–288, 1996.

[13] S. L. Sporer and B. Schwandt, "Paraverbal indicators of deception: A meta-analytic synthesis," *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, vol. 20, no. 4, pp. 421–446, 2006.

[14] S. A. Spence, T. F. Farrow, A. E. Herford, I. D. Wilkinson, Y. Zheng, and P. W. Woodruff, "Behavioural and functional anatomical correlates of deception in humans," *Neuroreport*, vol. 12, no. 13, pp. 2849–2853, 2001.

[15] L. Caso, A. Gnisci, A. Vrij, and S. Mann, "Processes underlying deception: An empirical analysis of truth and lies when manipulating the stakes," *Journal of Investigative Psychology and Offender Profiling*, vol. 2, no. 3, pp. 195–202, 2005.

[16] G. L. Lancaster, A. Vrij, L. Hope, and B. Waller, "Sorting the liars from the truth tellers: The benefits of asking unanticipated questions on lie detection," *Applied Cognitive Psychology*, vol. 27, no. 1, pp. 107–114, 2013.

[17] N. Mekkes, M. Groot, S. Wehrens, E. Hoekstra, M. K. Herbert, M. Brummer, D. Wever, N. N. D. Consortium, B. J. Eggen, A. Rozemuller *et al.*, "Natural language processing and modeling of clinical disease trajectories across brain disorders," *medRxiv*, pp. 2022–09, 2022.

[18] R. A. Calvo, D. N. Milne, M. S. Hussain, and H. Christensen, "Natural language processing in mental health applications using non-clinical texts," *Natural Language Engineering*, vol. 23, no. 5, pp. 649–685, 2017.

[19] P. A. Pérez-Toro, J. C. Vásquez-Correa, M. Strauss, J. R. Orozco-Arroyave, and E. Nöth, "Natural language analysis to detect parkinson's disease," in *Text, Speech, and Dialogue: 22nd International Conference, TSD 2019, Ljubljana, Slovenia, September 11–13, 2019, Proceedings 22*. Springer, 2019, pp. 82–90.

[20] G. Notelaers and S. Einarsen, "The world turns at 33 and 45: Defining simple cutoff scores for the negative acts questionnaire–revised in a representative sample," *European Journal of Work and Organizational Psychology*, vol. 22, no. 6, pp. 670–682, 2013.

[21] M. Shevlin, P. Hyland, M. Ben-Ezra, T. Karatzias, M. Cloitre, F. Vallières, R. Bachem, and A. Maercker, "Measuring icd-11 adjustment disorder: The development and initial validation of the international adjustment disorder questionnaire," *Acta Psychiatrica Scandinavica*, vol. 141, no. 3, pp. 265–274, 2020.

[22] A. P. Association *et al.*, "Inventario di personalità per il dsm-5—versione breve (pid-5-bf)—adulto," *Scale di Valutazione PID-5 ADULTI; Fossati, A., Borroni, S., Eds*, 2015.

[23] C. A. Blevins, F. W. Weathers, M. T. Davis, T. K. Witte, and J. L. Domino, "The posttraumatic stress disorder checklist for dsm-5 (pcl-5): Development and initial psychometric evaluation," *Journal of traumatic stress*, vol. 28, no. 6, pp. 489–498, 2015.

[24] E. Sanavio, "Le scale cba," *Milano, Cortina Editore*, 2002.

[25] D. S. Weiss, "The impact of event scale: revised," *Cross-cultural assessment of psychological trauma and PTSD*, pp. 219–238, 2007.

[26] G. Smith, S. Del Sala, R. H. Logie, and E. A. Maylor, "Prospective and retrospective memory in normal ageing and dementia: A questionnaire study," *Memory*, vol. 8, no. 5, pp. 311–321, 2000.

[27] K. Kroenke, R. L. Spitzer, and J. B. Williams, "The phq-9: validity of a brief depression severity measure," *Journal of general internal medicine*, vol. 16, no. 9, pp. 606–613, 2001.

[28] R. L. Spitzer, K. Kroenke, J. B. Williams, and B. Löwe, "A brief measure for assessing generalized anxiety disorder: the gad-7," *Archives of internal medicine*, vol. 166, no. 10, pp. 1092–1097, 2006.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] A. Zhang, Z. C. Lipton, M. Li, and A. J. Smola, "Dive into deep learning," *arXiv preprint arXiv:2106.11342*, 2021.

[31] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[32] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[33] L. Parisi, S. Francia, and P. Magnani, "Umberto: an italian language model trained with whole word masking," *Original-date*, vol. 55, p. 31Z, 2020.

[34] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[35] F. Bianchi, D. Nozza, D. Hovy *et al.*, "Feel-it: Emotion and sentiment classification for the italian language," in *Proceedings of the Eleventh Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Association for Computational Linguistics, 2021.

[36] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[37] A. M. Lamb, A. G. ALIAS PARTH GOYAL, Y. Zhang, S. Zhang, A. C. Courville, and Y. Bengio, "Professor forcing: A new algorithm for training recurrent networks," *Advances in neural information processing systems*, vol. 29, 2016.

[38] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, E. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *arXiv preprint arXiv:2210.11416*, 2022.

[39] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. Mooney, and K. Saenko, "Translating videos to natural language using deep recurrent neural networks," *arXiv preprint arXiv:1412.4729*, 2014.

[40] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, "Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning," *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[41] Q. Yang, Y. Zhang, W. Dai, and S. J. Pan, *Transfer learning*. Cambridge University Press, 2020.

[42] R. Caruana, *Multitask learning*. Springer, 1998.

[43] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in neural information processing systems*, vol. 27, 2014.

[44] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[45] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

[46] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[47] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," in *International conference on machine learning*. PMLR, 2014, pp. 647–655.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[49] B. Zoph, D. Yuret, J. May, and K. Knight, "Transfer learning for low-resource neural machine translation," *arXiv preprint arXiv:1604.02201*, 2016.

[50] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado *et al.*, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[51] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[52] P. Mishra and K. Sarawadekar, "Polynomial learning rate policy with warm restart for deep neural network," in *TENCON 2019-2019 IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 2087–2092.

[53] A. G. Thalmayer, G. Saucier, and A. Eigenhuis, "Comparative validity of brief to medium-length big five and big six personality questionnaires." *Psychological assessment*, vol. 23, no. 4, p. 995, 2011.

[54] A. K. Vijayakumar, M. Cogswell, R. R. Selvaraju, Q. Sun, S. Lee, D. Crandall, and D. Batra, "Diverse beam search: Decoding diverse solutions from neural sequence models," *arXiv preprint arXiv:1610.02424*, 2016.

Acknowledgments

I would like to thank my advisor Prof. Sartori for transmitting his passion for the topic of Lie Detection and guiding me through the development of this work, as well as being a wonderful teacher of the subject of psychology and applied machine learning to psychology, pointing me to many indispensable sources of knowledge. I would like also to thank Giulia Melis and Riccardo Lo Conte for suggesting me numerous resources to write my brief introduction to the matter of deception. I would like to thank my colleagues at the university for sharing this path with me, in a mutual exchange of knowledge and help. My thanksgivings are due also to all the Professors of the Master's Degree course in Data Science at the University of Padua, for illuminating me on this fascinating world of Data Science, allowing me to be, even though yet with very modest capabilities, a data scientist today. Thanks for transmitting to me the knowledge and the passion, the motivation, also through some failed exams. I like to think that this is just the beginning of my learning, and I would love to go deeper in every subject I had the opportunity to learn in this Master, and I am aware that I could never do it if it was not for all the professor that I met here. I would also like to thank the Data Science community in the largest sense, for contributing every day to share knowledge, as I drew countless times from it. I would like to thank also my friends, the old and the new ones, whose value to me cannot be summarized here without writing an inappropriate number of pages for this context. I would limit to say that good friends are a beautiful luxury in times of joy and invaluable help during difficult times. From the deep of my heart, I would like to thank my family for supporting me, in this case, I can summarize here what are they to me: something so essential like the food and the water that I eat and drink every day. I would not be the person that I am if they were not what they have been and continue to be for me. One special regard goes to my father, who is not with us anymore, but sure will live in my heart and in my actions. He taught me, by being the example, the most precious values that I live by, and I could not be me without them. Finally, I would like to thank all the people, alive or not, who have inspired me with their lives as I learned so much from observing what they have done, without the possibility to repay them. I would like to conclude this acknowledgment, by thanking whoever will take the time to read this work, I hope it helps you in what you are doing or simply fascinates you about Data Science halfway how I am fascinated about it. If you have any comments or suggestions, please write me to roberto98russo@gmail.com.