DEPARTMENT OF INFORMATICS

TECHNICAL UNIVERSITY OF MUNICH AND STANFORD UNIVERSITY

Master's Thesis in Robotics, Cognition, Intelligence

Explaining Neural NLP Models To Understand Students' Career Choices

Katharina Hermann

DEPARTMENT OF INFORMATICS

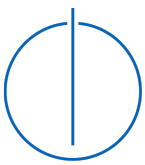
TECHNICAL UNIVERSITY OF MUNICH AND STANFORD UNIVERSITY

Master's Thesis in Robotics, Cognition, Intelligence

Explaining Neural NLP Models To Understand Students' Career Choices

Erklärung von neuronalen NLP-Modellen zum Verständnis der Berufswahl von Studenten

Author:	Katharina Hermann
Supervisor:	Prof. Georg Groh, Prof. Sheri Sheppard
Advisor:	Edoardo Mosca (M.Sc.)
Submission Date:	12/15/2021



I confirm that this master's thesis in robotics, cognition, intelligence is my own work and I have documented all sources and material used.

Munich, 12/15/2021 Katharina Hermann

Acknowledgments

I would first like to thank my supervisor from TUM, Edoardo Mosca, who supported and guided me in how to conduct the experiments in a systematic way in order to answer the research questions thoroughly.

Second, I would like to thank my supervisor from Stanford, Professor Sheri Sheppard, who gave me guidance in getting insights from a social science's perspective.

Third, I would like to thank my family and friends for giving me great emotional support – even remotely. I especially would like to mention the DEL 8 for being like a family at Stanford and encouraging me to think outside the box.

Abstract

Language is an elaborate construct for human communication. It transports explicit information as well as underlying social beliefs, norms, and individual thoughts. As such, analyzing language (e.g., survey answers) can generate great insights for researchers. The two dominant practices comprise traditional closed-vocabulary and open-vocabulary methods. Whereas the former introduces human bias and is resource-intensive, the latter overcomes these challenges with shallow *Natural Language Processing* (NLP) algorithms. Nonetheless, both methods fail to consider contextual information.

This thesis investigates survey answers from the EMS 1.0 dataset – using deep-learning-based methods from NLP and *eXplainable Artificial Intelligence* (XAI) – to answer the questions of whether neural networks can extract correlations from the survey answers at hand and which methods are suitable to make these correlations understandable for humans. This thesis proposes a general, new approach for survey analysis entailing mixed data: First, the deep neural model of this work predicts a variable of interest by simultaneously processing numerical and text inputs (from closed- and open-ended questions respectively). We use *Bidirectional Encoder Representations from Transformers* (BERT) to extract contextual correlations from the text with high precision compared to traditional methods. Previously, deep models could not be applied to such correlation analysis due to their black box characteristics. The new research field of XAI – introducing methods to explain model correlations – enables us now to use deep models for survey analysis. In particular, *SHapley Additive exPlanations* (SHAP) allows us to measure the impact of inputs on a model's prediction. This thesis tests two different variations of SHAP – classic SHAP methods on a lower-feature level and *ConceptSHAP* on a higher-conceptual level – to generate a holistic understanding of the model's extracted correlations between the survey's variables.

This approach is applied to the EMS 1.0 dataset studying influencing factors affecting a student's career goal and delivers promising results: First, we identified the most important numerical factors affecting a student's career goal, with high precision. Second, we revealed influencing factors not yet identified for career goals by studying the text answers related to the career goal both on a lower input and higher concept level. This thesis proposes the model at hand as a suitable tool in order to analyze students' survey answers and encourages broader application of the approach to survey-based data inquiries.

Contents

1. Introduction and Motivation

"Language is the principal means whereby we conduct our social lives." (Kramsch and Widdowson 1998, p. 3). It is the funnel through which humans process impressions, feelings, or intentions revealing insights into a person's thinking (Roberts, Stewart, Tingley, et al. 2014). Hence, analyzing language or text is of great value for answering relevant research questions, especially for sociological or psychological applications (Vaske 2019). This topic of discussion has permeated the academic sphere dating back to the time of Freud (1938), who introduced the concept of an individuals' spoken mistakes as indicators of hidden intentions - the Freudian Slip.

One primary research methodology for investigating research questions in social sciences is survey analysis as taught by Vaske (2019). These surveys can either be based on open-ended questions or closed-ended questions. Open-ended questions collect a person's free thoughts in the form of qualitative text, while for closed-ended questions, the answer choice is specified by the researcher. Therefore, closed-ended questions with numerical answers can be analyzed quantitatively to deliver clear statistical answers to a research question. However, they produce limited insights due to a lack of detailed information compared to open-ended answers.

Unlike closed-ended survey answers, language in open-ended answers conveys additional unintentional information that cannot be expressed explicitly (Kramsch and Widdowson 1998). Hence, open-ended answers provide more extensive insights into a survey's subject of interest, including its thinking or beliefs, personal or societal context, personality traits, and former experiences (Roberts, Stewart, Tingley, et al. 2014; Katz, Norris, Alsharif, et al. 2021).

Motivated by this potential, researchers for qualitative analysis of open-ended answers manually extract topics or codes in a text, grouping co-occurring words with techniques like open-coding (Levine, Björklund, S. Gilmartin, and Sheppard 2017; Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken 2016). These topics or qualitative codes are used to further *explain* other variables or draw rational conclusions from their structure (Eichstaedt, Kern, Yaden, et al. 2020). When applying this research methodology, it is crucial for a qualitative coding scheme (being a concept for characterizing a subset of the text data) and its correlations to a variable of interest to be explainable. However, assigning qualitative codes manually is a very slow and expensive process (Leeson, Resnick, Alexander, and Rovers 2019). Moreover,

it carries the risk of human bias (Katz, Norris, Alsharif, et al. 2021; LeCompte 2000).

Therefore methods from *Natural Language Processing* (NLP), mainly based on *Bag-Of-Words* (BOW), represent a viable and alternative methodology by automatically extracting value out of text (Eichstaedt, Kern, Yaden, et al. 2020). These methods help to analyze text faster and with less bias while still being explainable (Eichstaedt, Kern, Yaden, et al. 2020). However, they rather extract low-level information from words in a text without taking the words' context into account. This limits the quality of the analysis.

This is where deep neural networks and especially language models such as transformers (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019; Vaswani, Shazeer, Parmar, et al. 2017; Radford, Wu, Child, et al. 2019) are particularly valuable. They can extract in-depth contextual information by modeling complex relations in unstructured text data with high accuracy. Researchers could generate more comprehensive insights from text if these models were transparent and explainable. This is characterized by the measurable predictive accuracy of shallow versus deep models (Eichstaedt, Kern, Yaden, et al. 2020; Murdoch, C. Singh, Kumbier, et al. 2019). However, the predictive accuracy of complex black box models often conflicts with so-called *explainability*, impeding their usage in various fields, where transparency is of the highest priority in terms of research reliability and validity (Murdoch, C. Singh, Kumbier, et al. 2019).

In order to address these obstacles and to explain the rationale of neural networks, the relatively new research strand of *eXplainbable Artificial Intelligence* (XAI) has emerged in recent years. It opens up new opportunities for applying neural language models to survey analysis (Murdoch, C. Singh, Kumbier, et al. 2019). In this thesis, we focus on combining the latest achievements in the fields of NLP based on *Deep Neural Networks* (DNNs) with the latest developments in the field of XAI in order to compensate for the lack of black box *explainability* and to offer a new approach for analyzing surveys containing open-ended questions faster, deeper and with less human bias.

While our approach can be used in multi-fold research settings and survey analyses, we specifically apply it to a dataset, called the *Engineering Major Survey* (EMS 1.0) (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017). It was conducted in 2015 at 27 universities in the US, examining engineering students' career choices. We have chosen this dataset since the extraction of driving factors for engineering students' career choices is relevant for deriving implications around social stability, economic growth, and entrepreneurship (Eesley and Y. Wang 2017). For instance, previous research about the entrepreneurial aspirations of students has been dominated mainly by the impact of external and predetermined social factors. They include demography, financial stability, social networks, or access to training and experiences (Eesley and Y. Wang 2017). As stated above and in contrast to previous research in that field, language is a great medium to also study underlying social beliefs. By analyzing open-ended questions

from the EMS 1.0 survey (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017), this thesis therefore offers a new approach to exploring implicit factors for engineering students' career decisions.

Beyond that, we bring forward a methodology for XAI- and NLP-based survey analysis that exploits the potential of open-ended survey questions by overcoming the limitations of traditional qualitative text analysis methods, as introduced above. Firstly, we want to prove that deep natural language models are able to successfully and efficiently extract complex relationships from survey answers containing open- and closed-ended answers. Secondly, we want to reveal those complex relationships with explainability methods to make them accessible and understandable for humans by extracting the most important features and latent topics or concepts. In order to achieve this goal, we answer the following two research questions:

1. RQ1: Can Neural Networks with Language Models extract in-depth relationships between variables from survey data with open-ended and closed-ended survey questions in order to predict the main variable of interest - an engineering student's career goal?

2. RQ2: Can we leverage explainability methods in order to first extract the functional influence of low-level numerical and word token features on the predictions and secondly extract latent concepts from the open-ended answers, revealing additional information about the individual survey participant?

In chapter 2 the relevant work for our research question is presented. As we are trying to get insights by combining different fields, this chapter describes the studies performed on the EMS 1.0 dataset (see section 2.1), traditional methods for qualitative analysis of open-ended survey questions (see section 2.2), methods for text classification with deep neural networks (see section 2.3) and explainability methods for these text classification models (see section 2.4). We also state the contribution of our work in this chapter (see section 2.5). In chapter 3 we give a detailed overview of the EMS 1.0 dataset, being our application basis for the methods used to answer the research questions. These methods are explained in chapter 4. This chapter is further divided into the classification of students' answers (see section 4.1) and the explanation of the corresponding predictions (see section 4.2) following the structure of the research questions. The same structure is used for chapter 5, where we present the results for both the classification model (see section 5.1) and the applied explainability methods (see section 5.2). The work is closed by a conclusion in chapter 6 and some further suggestions for future work in chapter 7.

2. Related Work

In this section we want to describe the relevant work from the four different fields, that confluence in this work: The studies analyzing students' career goals based on the EMS 1.0 survey dataset (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017), traditional methods for open-ended survey analysis, methods from deep learning for text classification and post-hoc explainability methods for deep neural black box models.

2.1. EMS Study and Students' Career Goals

A substantial amount of work already explores which factors influence career decisions, especially focusing on entrepreneurial and innovative behavior and mindset (Eesley and Y. Wang 2017; Lindquist, Sol, and Van Praag 2015).

As we are mainly interested in what drives this mindset for engineering students, the primary basis of this work are studies based on the Engineering Majors Survey (EMS). The EMS is a longitudinal U.S. nationwide survey initiated by the National Center for Engineering Pathways to Innovation (Epicenter) in 2015, designed to explore engineering students' career goals around entrepreneurship and innovation. It entails answers from students enrolled at 27 universities across the United States. The study is based on the *Social Cognitive Career Theory* (SCCT) from Lent, Brown, and Hackett (1994), an extensive framework describing the correlation of factors influencing a student's career goal as well as their *Innovation Self-Efficacy* (ISE) and *Engineering Task Self-Efficacy* (ETSE). These factors mainly describe students' experiences and their demographic background. Factors of similar categories like training and experiences or social influences and their effects on entrepreneurial aspirations have also been studied extensively in literature (S. Scott and Khurana 2003; Lazear 2004; Stuart and Ding 2006; Nanda and Sørensen 2010).

Working with an open-ended question from EMS 1.0 the study of Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken (2016) chose a qualitative approach to manually extract an essential set of variables that students consider when thinking about their career goal. The question asks to share future career plans for the next five years or beyond (Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken 2016). Based on this question, the authors identified three principal

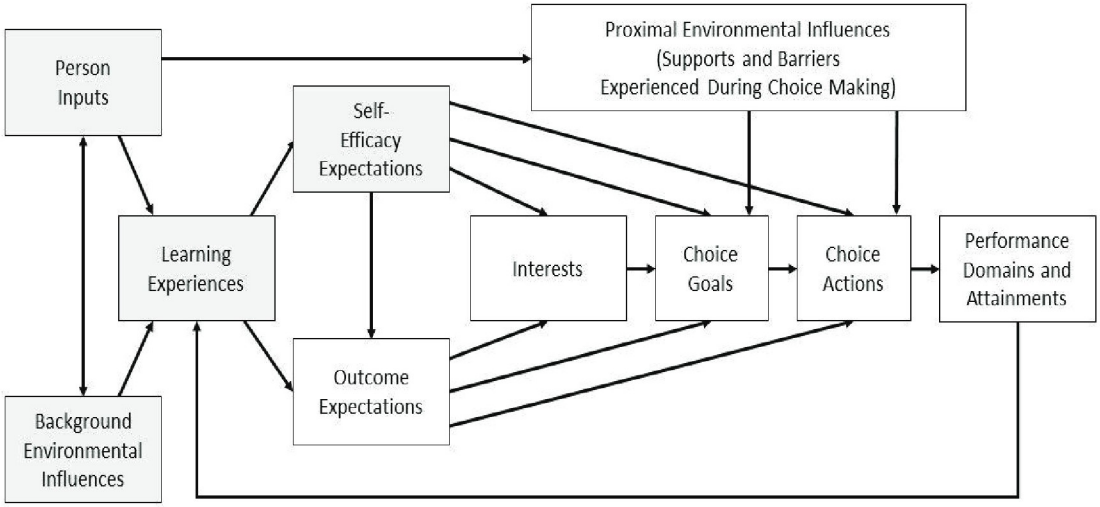


Figure 2.1.: Social Cognitive Career Theory (SCCT) model (Lent, Brown, and Hackett 1994, p. 93). Shaded nodes are included in the EMS study.(M. Schar, S. Gilmartin, Rieken, et al. 2017, p. 4)

codes or concepts that characterize how students think of their career goal (Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken 2016, p. 8):

1. "Overall plan", describing the clarity of plans and whether they involve engineering
2. "Career characteristics", giving details on the different career directions
3. "Career motivations", highlighting the drivers for the student's career goals

This study purely focuses on direct statements about which factors students consider when thinking of their career goals in detail. However, it does not investigate whether we can see a difference in the language used to express these career thoughts for different groups of students. This investigation in language could provide insight into behavioral, societal and personality differences that might influence thought processes (Katz, Norris, Alsharif, et al. 2021) - shedding light on the hidden "why".

There is a second study analyzing an open-ended question from EMS 1.0 Levine, Björklund, S. Gilmartin, and Sheppard (2017). This work examines the question of the EMS 1.0, asking about how the EMS 1.0 questions inspired students to think about their education or future differently. Similar to the work of Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken (2016), several variables were extracted as the first step with manual coding schemes from the answers in order to

statistically analyze those answers quantitatively as a second step. Some of the extracted variables from those answers are innovation and entrepreneurship that directly indicate the students' reflection on entrepreneurial motivation. However, the study does not investigate if and how those extracted factors correlate with a student's career goal.

Both studies above rely on manual methods for extracting the variables or coding schemes, coming with certain limitations further discussed in section 2.2.1.

Other studies on the EMS analyze quantitative correlations between numerical variables (Atwood, S. Gilmartin, Harris, and Sheppard 2021; M. Schar, S. Gilmartin, Rieken, et al. 2017). The study from Atwood, S. Gilmartin, Harris, and Sheppard (2021) relates SCCT topics as students' demographic characteristics, first-generation status, low-income family background with their college experiences. Further, this study also looks at a variable called *Engineering Task Self-Efficacy* (ETSE), which was found to be lowest for students of *first-generation* (FG) and *low-income* (LI). M. Schar, S. Gilmartin, Rieken, et al. (2017) studied influences from background and learning experiences on ETSE and *Innovation Self-Efficacy* (ISE) using different explainable regression models. Both studies rely on ETSE and ISE as indicators for someones' later career choice (Lent, Brown, and Hackett 1994; M. Schar, S. Gilmartin, Rieken, et al. 2017). However, as stated in M. Schar, S. Gilmartin, Rieken, et al. (2017), the ETSE only relies on five items defining a score from 1-4 (from M. Schar, S. Gilmartin, Harris, et al. (2017)), and the ISE is measured only by innovative activities, limiting both variables as measurements for career goals (M. Schar, S. Gilmartin, Rieken, et al. 2017). Beyond that, the dependent variables ETSE and ISE have only been set in context to the background and experience-based variables. However, it is essential to look at a broader scope of influential factors for innovation and entrepreneurship (M. Schar, S. Gilmartin, Rieken, et al. 2017), which we try to find in this work by studying the influence of all variables on the career goal.

In summary, the studies described above examine questions surrounding a student's career goal, either by looking at how such a goal can look in detail (Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken 2016), whether students reflect about entrepreneurship (Levine, Björklund, S. Gilmartin, and Sheppard 2017), or which demographic characteristics correlate with each other and with the ETSE measurement (Atwood, S. Gilmartin, Harris, and Sheppard 2021). However, these studies do not exhaustively study the influences for career goals as they only choose to examine a subset of the questions asked by the EMS 1.0 study. Furthermore, they do not study implicit motivational factors for students' career goals extracted from the patterns in language used by different student groups in the study. While those factors could be connected to the other variables queried in the survey (from the SCCT framework of Lent, Brown, and Hackett (1994)), like experiences or demographic characteristics, they could also reveal deeper insights about a student's personality traits. There are several studies that highlight personality traits as the main driver for entrepreneurship and

innovation (Eesley and Y. Wang 2017; Brandstätter 2011; Obschonka and Stuetzer 2017). Most of them rely on manual coding schemes as well.

There is also a movement towards large-scale methods to analyze the substance of information that natural language gives about a person's entrepreneurial behavior or mindset (Obschonka, N. Lee, Rodríguez-Pose, et al. 2018). It studies whether natural language from social media encodes information that can be extracted with large-scale methods like LDA topic modeling (see section 2.2.2) to predict an entrepreneurial personality. In their study, an entrepreneurial personality is characterized by the Big Five personality traits - extraversion, conscientiousness, neuroticism, openness, and agreeableness. However, they do not study characteristics in language *explaining* that prediction.

2.2. Qualitative Analysis of Open-ended Survey Questions

Methods for qualitative text analysis play a considerable role in psychological and sociological studies as they have several advantages to quantitative analysis of closed-ended questions (Roberts, Stewart, Tingley, et al. 2014; Katz, Norris, Alsharif, et al. 2021). They allow to extract information from language revealing people's emotions and thinking, as well as distilling attitudes salient beyond the time of writing an answer depending on people's social and cultural background or education (Roberts, Stewart, Tingley, et al. 2014; Kramsch and Widdowson 1998).

Therefore, analyzing language has a long history in social sciences. The most prominent approach, also used by the works related to EMS 1.0 (see section 2.1) is open-coding following mostly the idea of *Grounded Theory Method* (GTM) (Bryant and Charmaz 2007). As elaborated by Bryant and Charmaz (2007) GTM is an inductive approach for building theories based on data in contrast to theory-based approaches, which try to falsify a hypothesis. Open-coding is the core analytic process for GTM by which concepts (codes) are manually extracted from qualitative data. While this approach is used widely, it is very resource-intensive and introduces human-bias (Bryant and Charmaz 2007).

One improvement over manual open-coding is automated text analysis methods. We consider two different streams in today's literature. One stream is theory-based using closed-vocabulary methods from social sciences. The other one similar to open-coding is data-driven, however using automated open-vocabulary methods from computer science. See Eichstaedt, Kern, Yaden, et al. (2020) for a detailed comparison of methods. Closed-vocabulary approaches have a stronger focus on *how* people think, while open-vocabulary approaches are interested in *what* people think (Eichstaedt, Kern, Yaden, et al. 2020).

While they are different in how they extract information, they have the same goal: Grouping co-occurring words in a document together to so-called dictionaries (for closed-vocabulary) or topics (for open-vocabulary). These dictionaries or topics are then used further to characterize and statistically analyze the text documents, for example, by making predictions of other variables with simple machine learning models (Eichstaedt, Kern, Yaden, et al. 2020). In the following and for our purpose, we unify the two terms dictionaries and topics under the term of concepts. By extracting concepts, we can reveal correlations between them and the predicted variables. A simple analysis of the weights of a linear model, for example, highlights the topics/dictionaries with the highest correlation to a prediction (Eichstaedt, Kern, Yaden, et al. 2020). This analysis is already a simple gradient-based explainability method (see chapter 2.4). For our work, these methods are relevant, as we are also interested in correlations between text concepts and the predicted career goal variable. However, while those methods follow a bottom-up approach by first extracting concepts and then predicting the career goal with shallow NLP methods, we take a top-down approach using contextualized word embeddings from BERT (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019) to predict the career goal and then extracting concepts based on the prediction. Contextualized word embeddings will be further explained in chapter 2.3.

In the following, we explain the two different approaches of open-and closed-vocabulary with their most important methods and their implications for our work.

2.2.1. Closed-Vocabulary Methods from Social Sciences

Automatized text analysis started with closed-vocabulary methods. They work with dictionaries, which are theoretically-derived word lists representing psychologically relevant categories from social sciences. For analyzing a document, one then just assigns words in a document to one of the dictionaries and computes the relative frequency of that *dic*tionary compared to the overall frequency of all dictionaries occurring in one *doc*ument. This conditional probability $p(dic|doc)$ is then used as a feature variable for further analysis like prediction (Eichstaedt, Kern, Yaden, et al. 2020).

There are different programs with specific dictionaries in literature, each derived for a specific application. The most prominent programs covering large content are the *General Inquirer* (GI; Stone, Bales, Namenwirth, and Ogilvie (1962)) with 182 dictionaries, DICTION (Hart 1984) with 31 dictionaries and *Linguistic Inquiry and Word Count* (LIWC; Pennebaker, Francis, and Booth (1999); Pennebaker, Booth, Boyd, and Francis (2015)) with 73 dictionaries. LIWC is the most widespread program used, promoting a distinction between function words (pronouns, articles, prepositions, and conjunctions) and content words. While less than 200 distinct function words make up half of the words used in a text, making them more efficient in statistical use than content words,

they can also reveal underlying psychological processes as they are used unconsciously (Mehl, Gosling, and Pennebaker 2006).

However, the use of dictionaries has strong limitations as it only focuses on the appearance of words without capturing contextual meaning. Furthermore, the list of keywords in a dictionary is hand-crafted which introduces human bias and lacks exhaustiveness (Renz, Carrington, and Badger 2018). Beyond these quality limitations, the whole process is time- and labor-intensive (Roberts, Stewart, Tingley, et al. 2014).

2.2.2. Open-Vocabulary Methods from Computer Science

Open-vocabulary methods have emerged as an automatized improvement to manual open-coding with the rising amount of available text data (Eichstaedt, Kern, Yaden, et al. 2020). Following also the GTM idea (Bryant and Charmaz 2007), they allow discovering concepts (topics) from data directly rather than theoretically construct them as dictionaries as for the closed-vocabulary approach (Roberts, Stewart, Tingley, et al. 2014). These topics can also be combined with closed-ended survey responses to use further background information as in the study of Roberts, Stewart, Tingley, et al. (2014). The main assumption behind those methods performing so-called topic modeling is that each document with words consists of not directly observable topics, and each topic is a collection of words (Eichstaedt, Kern, Yaden, et al. 2020).

Topic modeling is, therefore, equivalent to a "clustering" of words that co-occur in text, reducing the sparse high dimensional word-document space to a dense lower-dimensional word-topic and topic-document space (Guetterman, T. Chang, DeJonckheere, et al. 2018; Eichstaedt, Kern, Yaden, et al. 2020). Since these lower-dimensional topics are only represented implicitly by a collection of words belonging to them, they are called latent topics (Eichstaedt, Kern, Yaden, et al. 2020). These latent topics or micro-dictionaries already encode higher abstractions of language. That makes them first more suitable for further predictions than shallow word vectors and secondly enable to interpret the prediction, which is one of the most critical criteria in psychology (Eichstaedt, Kern, Yaden, et al. 2020). There are various methods to extract these latent topics, presented in the following.

One early approach for topic modeling is the *Latent Semantic Analysis (LSA)* from Deerwester, Dumais, Furnas, et al. (1990). It starts with a word-document matrix $A \in^{N \times M}$, counting the frequencies of each word n occurring in a document m, or a TF-IDF score matrix (Jones 1972) and performs a matrix factorization using a truncated Singular Value Decomposition (SVD):

$$A \approx USV^T \tag{2.1}$$

with $U \in^{N \times T}$ being the word-topic matrix with the truncated topic dimension and

$S \in^{T \times T}$ and $V \in^{M \times T}$ being the topic-document matrix. With these matrices, one can now compute similarities of different documents or words with measures such as cosine similarity (Eichstaedt, Kern, Yaden, et al. 2020). While this approach is quite efficient to use, Eichstaedt, Kern, Yaden, et al. (2020) mention several drawbacks. First, it lacks interpretability since we cannot directly assign words to topics or topics to documents due to arbitrary positive and negative matrix entries. Secondly, it ignores that words have multiple meanings. The reason is that SVD imposes mathematical constraints, modeling language as a global geometric space, which does not follow the way language is used.

The most prevalent approach for topic modeling is *Latent Dirichlet Allocation (LDA)* (Blei, Ng, and Jordan 2003), which uses a probabilistic Bayesian approach instead of SVD. As explained by Eichstaedt, Kern, Yaden, et al. (2020) LDA again starts with a word-document matrix. Opposed to LSA, it now fits a probabilistic model with two Dirichlet distributions for words over topics $p(w|t)$ and for topics over documents $p(t|d)$. Factor analysis identifies the parameters of the two distributions that best generate the data we observe in the word-document matrix. The distribution $p(t|d)$, providing the probability of a topic t occurring in document m, is now human-interpretable and can be used as features for further statistical analysis as studied, e.g., by Jayaratne and Jayatilleke (2020) for the prediction of personality traits. The number of topics (latent dimension) is non-trivial and must be set as a priory. The number of topics influences, for example, whether the model is fine-grained enough to differentiate word senses by assigning them to different topics (Eichstaedt, Kern, Yaden, et al. 2020). Like closed-vocabulary approaches, LDA has the advantage that the process of topic generation (topic modeling) is independent of the text and can be performed on a different dataset than topic extraction. For the application one only needs the assignment of words to topics with the distribution $p(t|w)$ to compute the desired distribution $p(t|d)$ as the following: $p(t|d) = \sum_{words \in topics} p(t|w) \cdot p(w|d)$.

Both LSA and LDA have the disadvantage of being bag-of-word models, purely counting word frequencies and neglecting the sequence of words. They, therefore, can only extract the meaning of a text to a certain limit, having no deep contextual understanding of the content. Furthermore, for analyzing correlations between text and an independent variable, these methods only explain a topic- and not on a word level. This limits the depth of the explanation.

While topic models extract meaning from a text on a document-topic level, we will also learn about word embeddings (see section 2.3) extracting meaning from a text on a word level by capturing a word's embedding within other words (Eichstaedt, Kern, Yaden, et al. 2020). Beyond typical NLP tasks like text classification (discussed in 2.3), those embeddings can also be helpful for psychological applications. Studying the semantic similarity of words by analyzing their distance in a general embedding space,

one can extract global associations (similarities) of concepts, which is referable to how human minds process information Bhatia (2017). By clustering semantic distances for adjectives Parrigon, Woo, Tay, and T. Wang (2017), for example, extracted a situation taxonomy.

2.3. Text Classification with Deep Neural Networks

In this work, we focus on deep NLP surrounding text classification and text understanding, being referred to as *Natural Language Understanding* (NLU).

2.3.1. Word Embeddings for Neural Architectures

Contrary to tabular or image data, for processing text data with a deep learning model, we first have to convert unstructured text sequences into a structured feature space of numerical vectors before feeding it into the network. There are many traditional techniques used in combination with simple *Machine Learning* (ML) algorithms like *Bag-Of-Words* (BOW; Zhang, Jin, and Zhou (2010)), word-document matrix (used also for LDA and LSA explained in section 2.2.2) and the improved method of *Term Frequency - Inverse Document Frequency* (TF-IDF; Jones (1972)) to represent words numerically. All those methods only capture the syntactic but not the semantic meaning of a text as they are based on the assumption that the order of words is not relevant (Z. Liu, Lin, and Sun 2020). Therefore, word embeddings extracted with methods like Word2Vec (Mikolov, K. Chen, Corrado, and Dean 2013) and GloVe (Pennington, Socher, and Manning n.d.) provide a significant improvement over those techniques, paving the way for neural networks for text data (Z. Liu, Lin, and Sun 2020). A word embedding "is a feature learning technique in which each word or phrase from the vocabulary is mapped to a [low dimensional] vector of real numbers" (Kowsari, Meimandi, Heidarysafa, et al. 2019, p. 7), which allows encoding the meaning of a word.

pre-trained word embeddings from Word2Vec (Mikolov, K. Chen, Corrado, and Dean 2013) and GloVe (Pennington, Socher, and Manning n.d.) have been widely used as fixed embedding vectors independently from the specific text data (Z. Liu, Lin, and Sun 2020) as input for neural NLP models. However, beyond using those fixed vectors, it is also possible to train the mapping of unstructured words to a numerical vector from scratch on the specific text.

However, embeddings trained with only one layer or received from Word2Vec (Mikolov, K. Chen, Corrado, and Dean 2013) or GloVe (Pennington, Socher, and Manning n.d.) captures only very low-level information of individual words. The initialization of models with shallow word embeddings, therefore, requires long training with large amounts of data (Ruder 2018).

2.3.2. Language Models and Transformer Architectures

Given the limitations of word embeddings, the trend has moved towards so-called (pre-trained) language models. They are deep networks that learn dynamical word representations to encode deeper semantic meaning and context information of larger text structures (Ruder 2018; Z. Liu, Lin, and Sun 2020). These language models (Peters, Neumann, Iyyer, et al. 2018; Howard and Ruder 2018; Devlin, M.-W. Chang, K. Lee, and Toutanova 2019) can be either fine-tuned or used in a direct pre-trained fashion - delivering high-level word and sequence representation vectors, which can be fed to subsequent multi-layer architectures equivalent to traditional fixed and low-level word embeddings.

In general, language models gain an understanding of language by being trained with the objective of next word prediction on a large corpus of text (Z. Liu, Lin, and Sun 2020). One of the first language models in 2018 was ELMo (Peters, Neumann, Iyyer, et al. 2018), being able to disambiguate words, delivering deep contextualized *Embeddings from Language MOdel*, contrary to Word2Vec (Mikolov, K. Chen, Corrado, and Dean 2013).

At the same time, the method of *Universal Language Model FIne-Tuning* (ULMFiT) (Howard and Ruder 2018) also based on an LSTM architecture, was a big advance for transfer learning in NLP, enabling the application of generic pre-trained language models to specific tasks in NLP - only by fine-tuning the language models instead of retraining entirely (Ruder 2018). The second revolution in NLP was transformer models with the paper *Attention is all you need* from Vaswani, Shazeer, Parmar, et al. (2017). Transformer models purely rely on a concept called *Attention* to deal with long-term dependencies in large word sequences. In combination with encoder-decoder structures they outpace LSTM models (Alammar 2018). While the original paper has focused on sequence-2-sequence models for language translation tasks, it laid the foundation for two important pre-trained language models: GPT-2 from Open-AI (Radford, Wu, Child, et al. 2019) for language generation and Google's *Bidirectional Encoder Representations from Transformers* (BERT) model (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019) for language understanding.

BERT was the first deeply bidirectional, unsupervised trained language representation model (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019). The training objective is to predict the next sentence and a word that has been masked before on a large text corpus with a vocabulary size of about 30k words (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019). BERT has paved the way for the most sophisticated open-source NLP models in research like the model RoBERTa from Y. Liu, Ott, Goyal, et al. (2019). As BERT outputs generic high-level word and sequence representations in the form of embedding vectors, the pre-trained model can be integrated and fine-tuned in any kind of model to perform

a large variety of NLP tasks like (in our case) text classification (see section 4.1.1). While the fine-tuning delivers very precise results, when performed for a large text corpus, the model can also be used frozen without fine-tuning as a "feature-based approach" (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019, p. 1), being a better option for small datasets. The context word and sequence embeddings from BERT, therefore, combine the advantage of capturing a deep semantic meaning of a text while still being applicable without fine-tuning or retraining like Word2Vec (Mikolov, K. Chen, Corrado, and Dean 2013) and GloVe (Pennington, Socher, and Manning n.d.) embeddings. This makes it the perfect model for our purpose of performing text classification on a small dataset.

BERT is very large and thus slow and costly. That is why smaller, more efficient versions like DistilBERT (Sanh, Debut, Chaumond, and Wolf 2019) have emerged. DistillBERT will be the model of choice for our work. This model (like BERT) takes word token embeddings (together with position and the segment embeddings) as an input, where each word (or chunk of a word) is represented with an integer (Sanh, Debut, Chaumond, and Wolf 2019). This encoding is performed with a pre-trained tokenizer. The model output is contextual word- and sequence-embeddings in the form of hidden state vectors ($\in \mathbb{R}^{768}$) each belonging to an input token. The sequence embedding belonging to a so-called CLS token (a special classification token in front of every sequence) captures the meaning of the whole sequence (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019).

2.4. Post-Hoc Explainability for NLP black box Models

Recent advances in NLP systems with deep architectures like transformers explained in the previous section 2.3 have led to enormous success in extracting complex relations in text, to interpret and classify it with surprising accuracy, not possible with traditional methods from NLP as discussed in section 2.2. However, architectures that capture such a high complexity with high accuracy come with the trade-off of being less interpretable - also referred to as black box models (Danilevsky, Qian, Aharonov, et al. n.d.; Murdoch, C. Singh, Kumbier, et al. 2019).

However, in many applications (as in medicine or the qualitative analysis of open-ended answers in psychology (see section 2.2)), we want to understand *why* a specific prediction is made. This is the reason for simple white-box models often being preferred over those complex backbox models (Murdoch, C. Singh, Kumbier, et al. 2019).

To address this interpretability-accuracy trade-off, the research field of *EXplainable Artificial Intelligence* (XAI) has emerged, proposing various methods for interpreting the prediction of complex models. The high abundance of methods has led to various

works (Murdoch, C. Singh, Kumbier, et al. 2019; Arrieta, Díaz-Rodríguez, Del Ser, et al. 2019; Atanasova, Simonsen, Lioma, and Augenstein 2020; Danilevsky, Qian, Aharonov, et al. n.d.; Guidotti, Monreale, Ruggieri, et al. 2018; Mathews 2019; Tjoa and Guan 2019; Molnar 2019) trying to find a consistent terminology and taxonomy for this still relatively young research field.

Murdoch, C. Singh, Kumbier, et al. (2019) define XAI (exchangeable with explainable, interpretable, or transparent ML) "as the extraction of relevant knowledge from a machine learning model concerning relationships either contained in data or learned by the model" (Murdoch, C. Singh, Kumbier, et al. 2019, p. 1).

2.4.1. Goals for Explainability and Taxonomy

We want to characterize methods depending on:

1. The explanation scope (what to explain)
2. The result of explanations (how to explain) - combining taxonomies from Ghorbani and J. Zou (2020) and Sajjad, Kokhlikyan, Dalvi, and Durrani (2021).

For the 1st dimension of the explanation scope we follow the distinction from Sajjad, Kokhlikyan, Dalvi, and Durrani (2021) of attribution or causation analysis and concept-based or fine-grained analysis as shown in figure 2.2:

1. Attribution or causation analysis methods (yellow part in figure 2.2) want to understand how input features or neurons influence a specific prediction or set of predictions - being mostly local explanations (Molnar 2019).
2. Concept-based or fine-grained analysis (blue part in figure 2.2), on the other hand, wants to extract human-interpretable concepts, consisting of sets of (text) features from several input samples sharing the same activation for specific neurons and groups of neurons. Concept-based explanations are global explanations for how the model reasons and captures information (Molnar 2019).
3. Global explanations (green part in figure 2.2), summarizing the general influence of input features for a whole class or general prediction head of the model, bridge the gap between causation and concept-based analysis: They analyze the causal effect but also summarize the set of features (into a concept) most important for the activation of the (one output) neuron. However, in the following, we declare it as global causation analysis as we see concepts more fine-grained - explaining inner elements (neurons) of a model.

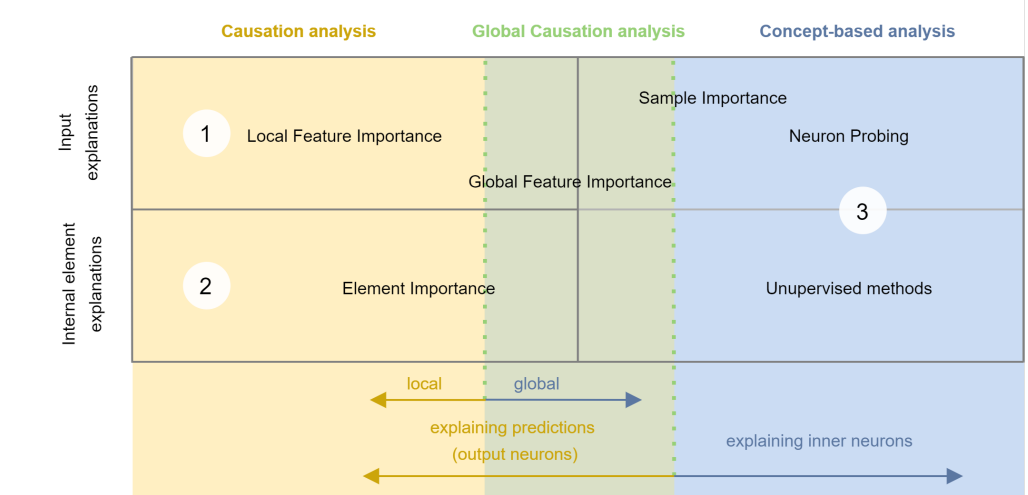


Figure 2.2.: Explainability methods classified by a taxonomy following Ghorbani and J. Zou (2020) and Sajjad, Kokhlikyan, Dalvi, and Durrani (2021). The yellow field describes causation analysis methods, while the blue field describes concept-based analysis methods. The intersection (green) of both methods are global causation analysis methods

For the second dimension (as shown on the y-axis of figure 2.2) - the result of explanation - we follow Ghorbani and J. Zou (2020). We distinguish between delivering an explanation on an input level, explaining predictions or neuron activations with the specific data fed to the model, or giving explanations only with the internal workings of the model itself. The latter explains which neurons or layers have triggered a specific prediction or another neuron or layer - independent of the data fed to the model. Ghorbani and J. Zou (2020) introduce the three terms feature importance, element importance, and sample importance, which the two dimensions can classify according to figure 2.2. For concept-based analysis (3), we can further classify the methods of neuron probing and unsupervised methods introduced by Sajjad, Kokhlikyan, Dalvi, and Durrani (2021).

In this work feature importance (1 - explained in section 2.4.2) on the causation analysis side will be relevant as a first step for highlighting the most important numerical features and word tokens to our prediction globally across all student instances (section 4.2.1). Combining methods from element importance (2 - explained in section 2.4.4) with concept-based analysis methods (3 - explained in section 2.4.3) will then become relevant for the second step of the analysis (section 4.2.2) for extracting concepts and delivering global explanations on how these concepts' influence the model's predictions (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021). All three parts will be explained in further detail in the following sections.

Furthermore, we characterize methods based on the scope of models for which they can deliver an explanation, being either model-agnostic (applicable to all types of models) or model-specific (applicable to only one type of model). Model-agnostic methods often explain the model after being trained (post-hoc) on a data input (feature) level, not an internal model level. Model-specific methods, on the other hand, can be either post-hoc or ad-hoc (model-based (Murdoch, C. Singh, Kumbier, et al. 2019)) for intrinsically interpretable models (like linear regression) (Molnar 2019). Murdoch, C. Singh, Kumbier, et al. (2019) distinguish primarily between model-based and post-hoc explainability. Inherently interpretable but less expressive model-based methods include clustering, dimensionality reduction, matrix factorization, and dictionary learning (Murdoch, C. Singh, Kumbier, et al. 2019). They are the prevailing methods used so far in psychology for analyzing open-ended questions and have been discussed in section 2.2. For this work, we prefer post-hoc methods since they provide insights into the complex relations learned by black box neural language models, which we aim to use to process complex data like text (Murdoch, C. Singh, Kumbier, et al. 2019) with more depth - giving high predictive accuracy. We, therefore, focus on post-hoc methods in the following.

2.4.2. Feature Importance

Methods for feature importance (also called feature attribution (Olah, Mordvintsev, and Schubert 2017)) analyze the influence of an input feature on the model's prediction. Feature importance can be either local - highlighting the critical features for a single instance - or global - computing the importance for the whole dataset (Murdoch, C. Singh, Kumbier, et al. 2019). In general, for feature importance, we must distinguish whether we only account for the relative influence, which a change of an input feature has (being referred to as sensitivity), or for the actual influence of the feature (being referred to as saliency) (Ancona, Ceolini, Öztireli, and Gross 2019). Sensitivity only measures the relative importance of a feature since we do not know whether the actual feature will be high or low. In contrast, saliency measures the actual influence of the feature on the output, as we multiply the feature value with its relative change.

To compute either the saliency or sensitivity of a feature, many different approaches have been proposed. We distinguish between gradient-based methods (Simonyan, Vedaldi, and Zisserman 2013; Denil, Demiraj, and Freitas 2014; J. Li, X. Chen, Hovy, and Jurafsky 2015; Springenberg, Dosovitskiy, Brox, and Riedmiller 2014; Sundararajan, Taly, and Yan 2017), propagation-based methods (Bach, Binder, Montavon, et al. 2015; Arras, Montavon, Müller, and Samek 2017; Arras, Osman, Müller, and Samek 2019; Montavon, Samek, and Müller 2017; Poerner, Roth, and Schütze 2018a), simplification-based methods (Ribeiro, S. Singh, and Guestrin 2016) and perturbation-based methods

(J. Li, Monroe, and Jurafsky 2016; Zeiler and Fergus 2013; Sundararajan and Najmi n.d.; Shapley 2016). While these terms help us characterize methods, they can also belong to several categories, and the transitions are fluent. Examples are Integrated Gradients (Sundararajan, Taly, and Yan 2017) or *SHapley Additive exPlanations* (SHAP; Lundberg and S.-I. Lee (2017)). The authors of SHAP (Lundberg and S.-I. Lee 2017) try to unify several methods and categories with the idea of additive feature importance. We will explain SHAP values in further detail together with other related concepts (Ribeiro, S. Singh, and Guestrin 2016; Shapley 2016) in chapters 2.4.2, 2.4.2 and 2.4.2 since we use this method for the first explainability experiment (section 4.2.1) to study word and numerical feature level attributions.

LIME

We are going to describe LIME (Ribeiro, S. Singh, and Guestrin 2016) in more detail as it will be of high relevancy for the method of SHAP values Lundberg and S.-I. Lee (2017), which we are going to use as a base for our explainability experiments. We are following the explanation of the authors and the one of Lundberg and S.-I. Lee (2017). LIME belongs to the group of simplification-based methods for feature importance. It tries to create interpretable surrogate models of lower complexity than their original models, for which the influence of an input feature can be seen more easily.

Following this idea, LIME tries to find a simplified interpretable model $g(x)$ (also called explanation model) that explains the prediction $f(x)$, for a single input x, being, therefore, a local method. The method can be applied post-hoc to any model, so it is further model-agnostic.

$g(x)$ can be any model, but the simplest one is a linear regression model following equation 2.2 stating the term *Linear LIME* (Ribeiro, S. Singh, and Guestrin 2016).

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{2.2}$$

M is the number of features i. $x' \in 0, 1^M$ is a so called "interpretable input" (Ribeiro, S. Singh, and Guestrin 2016, p. 3) (or "simplified input" (Lundberg and S.-I. Lee 2017, p. 2)) vector, which indicates for the input of text tokens which tokens are present (with a 1) and which are not present (with a 0). This vector x' is mapped to the original input vector x with the mapping function $x = h_x(x')$. ϕ_i (weight $\times$ value) is therefore the overall contribution or importance of feature i to the prediction $g(x')$.

The one $g(x')$ with its corresponding parameters ϕ_i that best approximates the original model $f(h_x(x'))$ is found (from within all possible models G) by minimizing the loss L following equation 2.3. The loss is optimized over a set of perturbation

samples z' around x', weighted by the local kernel $\pi_{x'}$ indicating their distance from x' (Ribeiro, S. Singh, and Guestrin 2016).

$$minL(f, g(z'), \pi_{x'}) + \Omega(g) \tag{2.3}$$

$\Omega(g)$ describes the complexity of the model g, which in theory gets penalized, but must be chosen manually.

Shapley Values

Shapley values (Shapley 2016) is a relevant concept of feature importance also building the basis for SHAP values (Lundberg and S.-I. Lee 2017) and will be discussed in further detail in the following. It belongs to the group of perturbation-based methods.

Opposed to the gradient- or propagation-based methods, perturbation-based methods (J. Li, Monroe, and Jurafsky 2016; Zeiler and Fergus 2013; Shapley 2016) do not compute a forward pass and a backward pass to compute the feature importance directly. They instead create perturbed instances of input x by occluding, deleting, or modifying certain features of that input $x_{|x_i=0}$ and measure the change in prediction for those modified instances of x:

$$f_c(x) - f_c(x_{|x_i=0}) \tag{2.4}$$

While Zeiler and Fergus (2013) have studied occlusion for images, J. Li, Monroe, and Jurafsky (2016) have studied occlusion for text input.

Shapley values is an approach from Game Theory being the only method that satisfies several properties, together defining a fair distribution of feature attribution (Lundberg and S.-I. Lee 2017). Like in a game we use the term *players*, which corresponds to the features i of the input instance x, and *payout*, which corresponds to the model prediction $f(x)$. The goal of Shapley values is to determine the contribution of the *players* to the game (the process of prediction for a single instance) and distribute the *payout* fairly among them. The individual *payout* to each of the *players* (features i) is called its *Shapley Value* ϕ_i, which represents the feature's i importance or contribution to the prediction. If we compute the contribution of a feature i to the *payout* for one single instance, it is a local method. If we sum it over all instances, it becomes a global method.

The Shapley Value for the contribution (importance) of a feature i to a prediction, for instance, x can be computed as the following:

$$\phi_i(x, f) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!} [f_x(z') - f_x(Z' \setminus i)] \tag{2.5}$$

M is the number of features, in our case, the total vocabulary (number of tokens) size. x' is the "simplified input" (Lundberg and S.-I. Lee 2017, p. 2) vector equivalent to the one used in LIME (Ribeiro, S. Singh, and Guestrin 2016). z' are again the perturbation vectors of x', but as opposed to LIME, they are not random but represent a set of all possible sub-combinations of the non-zero features of the simplified input vector x'. This subset characteristic is expressed with $z' \subseteq x'$. $|z'|$ is the number of present (non-zero) features in z', $z' \setminus i$ represents the vector z' with feature i being set to zero.

$f_x(z') = f(h_x(z'))$, with $h_x(z')$ is the prediction of the model f for the simplified input perturbation z'. Intuitively, the Shapley Value equation 2.5 computes the change in model prediction for adding feature i to a current set of features z'. This difference is computed for all feature perturbation combinations z', weighted by a factor, which considers the present features with $|z'|!$ and which features are missing $(M - |z'| - 1)!$, normalized by the M the number of total features. Therefore, the overall influence of i is a fairly weighted sum of all the differences for every possible feature combination z'.

With this fair payout distribution, the Shapley value is the only feature attribution method that fulfills the properties *Efficiency, Symmetry, Dummy, Additivity* (Lundberg and S.-I. Lee 2017). As these properties are desirable, Shapley values have a strong legitimacy (Sundararajan and Najmi n.d.).

Overall said the Shapley value of a single feature value ϕ_i measures how much it contributes to the difference between the prediction y_x for the current set of feature values and the mean prediction of all samples X (Molnar 2019). Therefore, the. Summing the Shapley values ϕ_i for all features is, therefore, gives the difference of the prediction for x and the average prediction - being the property of *Efficiency*.

$$\sum_{i=1}^{M} \phi_i(x, f) = f(x) - E_X(f(X)) \tag{2.6}$$

However, computing Shapley values exactly is very inefficient, and the effort rises exponentially with the number of features, as the model must be re-trained for all feature combinations (Molnar 2019). This drawback has fostered methods, which compute Shapley values more efficiently, such as *BShap* (Sundararajan and Najmi n.d.) or *Integrated Gradients* (Sundararajan, Taly, and Yan 2017).

Beyond its inefficiency, Shapley values also have the disadvantage of delivering only an attribution value per feature and not an explanation model like LIME (Molnar 2019). SHAP values address this issue as presented in the next section 2.4.2. Further, Shapley values can also include unrealistic data samples by permuting x' to z', as those permutations are not from the real dataset and therefore not respecting correlating features (Molnar 2019).

Additive Feature Importance: SHAP

SHAP values are a unified measure of feature importance (Lundberg and S.-I. Lee 2017). The authors state that SHAP unifies several approaches like LIME, Shapley values, DeepLift, layer-relevance propagation, as they all adhere to the definition of the class of additive feature attribution methods proposed by the authors (see equation 2.7). According to this class of methods, we use an explanation model g that linearly sums up the effects (importance) ϕ_i attributed to each simplified input feature x' to explain the output $f(x)$ of the original model. For example. in comparing equation 2.7 to the equivalent equation 2.2 from section 2.4.2, we can show directly that linear LIME is an additive feature attribution method.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^{M} \phi_i x'_i \tag{2.7}$$

With ϕ_0 being the mean prediction for all samples $E_X(f(X))$, we can see that the sum of all feature attributions ϕ_i is equivalent to the difference of the actual prediction for a sample x to the mean predictions and ϕ_i the individual contribution of feature i - being equivalent to equation 2.6.

Lundberg and S.-I. Lee (2017) state that the class of additive feature attribution methods only have one unique solution for the feature importances ϕ_i, which fulfill the three desired properties of local accuracy, missingness and consistency. This solution are the SHAP values, which are Shapley values (Shapley 2016) for the conditional expectation model $f(h_x(z')) = f(z_S) = E[f(z)|z_S]$ with z_S having only non-zero values for a set S.

Computing SHAP values exactly is again very inefficient. That is why the paper from Lundberg and S.-I. Lee (2017) proposes methods like KernelSHAP, MaxSHAP, DeepSHAP to approximate SHAP values by combining insights from additive feature attribution methods like LIME and DeepLift with Shapley values (similar to Shapley value sampling but with different strategies than sampling).

One model-agnostic method of how to compute SHAP efficiently and with a good approximation is KernelSHAP (Lundberg and S.-I. Lee 2017). It is a combination of Shapley values and linear LIME. Since the high computational cost of Shapley values is one of their disadvantages, we choose the LIME approach (Ribeiro, S. Singh, and Guestrin 2016) for computing the feature importance ϕ_i as parameters of the linear explanation model g, by optimizing the loss equation 2.3. In particular, we choose model complexity $\Omega(g) = 0$ (linear model) and the weighting kernel $\pi_{z'}$ as well as the loss function $L(f, g, \pi_{z'})$, such that equation 2.7 is a solution to equation 2.3 with the Shapley values as its feature attribution parameters. In LIME, the weighting kernel $\pi_{z'}$ is computed differently by assigning low weights to far simplified inputs (few 1's),

while complexity and the loss function are chosen heuristically (Ribeiro, S. Singh, and Guestrin 2016) - which is why the solution to LIME does not reconstruct the Shapley values (Molnar 2019).

$$\pi_{z'} = \frac{M-1}{(M \, choose \, |z'|)|z'| \, (M-|z'|)} \tag{2.8}$$

$$L(f, g, \pi_{z'}) = \sum_{z' \in Z} [f(h_x^{-1}(z')) - g(z')]^2 \, \pi'_z \tag{2.9}$$

As the linear LIME model is a linear explanation model, we can solve equation 2.9 to compute all SHAP values jointly with a weighted linear regression.

SHAP combines the advantage of Shapley values, having a theoretical foundation in game theory (Molnar 2019), with distributing the *payout* fairly among its *players*, with the advantage of other methods like LIME. It is thus more efficient and easier to compute. Beyond that, the SHAP values for individual samples can be summed following equation 2.10, delivering a global model interpretation across all samples X, which is not possible in general for local methods like LIME (Setzu, Guidotti, Monreale, et al. 2021).

$$I_i = \sum_{n=1}^{N} |\phi_i^{(n)}| \tag{2.10}$$

This extension from local to global is only possible with Shapley values and would not be consistent for methods like LIME (Molnar 2019). Due to those advantages, SHAP values are a widespread and commonly used method in literature. Hence, they are the explainability method used for our first experiment in section 4.2.1.

Nevertheless, SHAP values still have some disadvantages: KernelSHAP is still relatively slow and ignores feature dependence (Molnar 2019). Alternatively, disadvantages inherent to Shapley values like the inclusion of unrealistic data instances when features are correlated (Molnar 2019).

2.4.3. Concept Analysis

Feature and element attribution or importance methods give, as a result, a summary of how features and elements influence the prediction of either an individual sample (local methods) or the prediction over a group or the whole set of samples (global methods). Nevertheless, a model can also be explained globally (for all samples) by showing which concepts are represented by one or several neurons (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021).

Concepts can be of various granularity, being either higher-level, like gender or religion, or summarizing lower-level concepts, like parts-of-speech (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021). While they always consist of several global features across samples, they can be represented in various formats generated differently.

Feature and Sample Importance

One way is using feature importance methods (J. Li, X. Chen, Hovy, and Jurafsky 2015; Karpathy, J. Johnson, and F.-F. Li 2015) explained in 2.4.2 to extract the most important features (or word tokens for language tasks) for a similar set of neurons (instead of the prediction), forming concepts bottom-up.

Another way of building concepts is sample importance (Ghorbani and J. Zou 2020) or corpus-selection (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021) showing representative examples for various sample features activating one or several neurons. Such methods are always global since the example represents a group or the whole set of data samples. As a result of this we distinguish between methods that take an example from the existing data set to make an explanation (Kádár, Chrupała, and Alishahi 2016; Koh and Liang n.d.) and methods that generate an explanation example (Na, Choe, D.-H. Lee, and G. Kim 2019; Erhan, Bengio, Courville, and Vincent 2009; Poerner, Roth, and Schütze 2018b).

Feature importance methods are primarily used in causation analysis, explaining predictions locally and globally rather than single neurons. Similar to that, sample importance methods can also explain predictions instead of neurons. Related works for taking examples from the dataset are (Aubakirova and Bansal 2016; Girshick, Donahue, Darrell, and Malik 2014), while (Simonyan, Vedaldi, and Zisserman 2013; Olah, Mordvintsev, and Schubert 2017; Nguyen, Yosinski, and Clune 2016; Nguyen, Dosovitskiy, Yosinski, et al. 2016; Mordvintsev, Olah, and Tyka 2015) focuses on generating examples. However, sample importance is only global since the example always represents a group or the whole set of data samples. Theoretically, suppose we explain the output neuron of a classification model by an (ex-)sample or a set of most important features globally over all samples. In that case, we can say that gives a very general explanation of the features forming one concept maximizing the activation of one of the output class neurons. This connects global explanations for predictions with concepts. However, these kinds of explanations are more global attribution analyses. We see concepts more in the narrow sense of representing neurons in the inner-working of a model to extract more fine-grained concepts (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021).

Neuron Probing

The third way of extracting concepts is called neuron probing methods by Sajjad, Kokhlikyan, Dalvi, and Durrani (2021). One manually designs and assigns concepts top-down to features building a supervised dataset. One then trains a second (linear or random forest) classifier taking neuron activations from the original model as input and the concept as the label. The most critical neurons for one concept are then derived by input saliency (Dalvi, Durrani, Sajjad, et al. 2018; Valipour, E.-S. Lee, Jamacaro, and Bessega 2019; Hewitt and Liang 2019).

Unsupervised methods

However, the methods described come with the trade-off of requiring a supervised dataset for every concept, which is why we introduce a last unsupervised type of method. These methods try to cluster neurons with similar activations and extract the most important neurons for the model (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021). We can again distinguish between neuron ablation methods (Lakretz, Kruszewski, Desbordes, et al. 2019; J. Li, Monroe, and Jurafsky 2016) connected to perturbation-based methods described in chapter 2.4.2, gaussian-based probing (Torroba Hennigen, Williams, and Cotterell 2020), matrix factorization (Olah, Satyanarayan, I. Johnson, et al. 2018) for which not much work has been done in NLP so far (Alammar 2020), activation- (Meyes, Puiseau, Posada-Moreno, and Meisen 2020) and correlation-based (Dalvi, Sajjad, Durrani, and Belinkov 2020) clustering methods and multi-model search (Bau*, Belinkov*, Sajjad, et al. 2019). While these clusters of neurons are good for redundancy analysis (Dalvi, Sajjad, Durrani, and Belinkov 2020) they still require further analysis to map them to concepts with a human in the loop or automatic concept annotation for single neurons, e.g., with the methods described above (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021).

2.4.4. Element Importance

As stated in chapter 1, our goal is to extract latent topics or concepts from the students' open-ended answers. Therefore, we are not only interested in word activation patterns from feature importance but also want to find out how neurons contribute to the model's predictions. Additionally, we want to interpret different clusters of activation patterns as topics/latent variables by methods from concept analysis (section 2.4.3).

Gopinath, Converse, Păsăreanu, and Taly (2019) and Sajjad, Kokhlikyan, Dalvi, and Durrani (2021) explicitly point out the difference between understanding the function of a hidden unit/neuron, which is studied intensively by concept-based analysis (see section 2.4.3) and identifying its importance for the overall prediction of an input x. This

input is referred to as attribution analysis by Sajjad, Kokhlikyan, Dalvi, and Durrani (2021). In this section, we focus on the latter. Gopinath, Converse, Păsăreanu, and Taly (2019) further introduce the term neuron *conductance* as equivalent to neuron importance extending the term of *attribution* used for features. For neuron importance, similar approaches to feature importance have been studied. Early works try simplification-based methods (similar to section 2.4.2) (Le 2013; Alain and Bengio 2016), leading to ambiguity to whether the neurons of the approximated model and the original model have the same influence (Gopinath, Converse, Păsăreanu, and Taly 2019). Others propose gradient-based methods (Gopinath, Converse, Păsăreanu, and Taly 2019).

While one might think of treating neurons in an embedding layer just like features in the input layer and applying the same methods to this layer (like saliency methods from Simonyan, Vedaldi, and Zisserman (2013), integrated gradients from Sundararajan, Taly, and Yan (2017)), this does not account for interactions of neurons between several layers (Ghorbani and J. Zou 2020). Beyond that, several approaches for understanding hidden units only work for a subset of hidden units, delivering no exhaustive investigation of neuron importance (Gopinath, Converse, Păsăreanu, and Taly 2019). We have methods for computing neuron importance, such as layer conductance (Gopinath, Converse, Păsăreanu, and Taly 2019) and Neuron Shapley (Ghorbani and J. Zou 2020)) and methods for neuron interactions, such as Archipelago (Tsang, Rambhatla, and Y. Liu 2020) or Shapley Taylor Interaction (Sundararajan, Dhamdhere, and Agarwal 2020). These methods are still quite new, opening up a high research opportunity for this work.

For our work, we are considering one other form of element importance, combining the approach of element importance with concept-based methods. Instead of extracting single or groups of neurons contributing to a model's prediction, we want to measure the contribution of neuron activation clusters (i.e., concepts) to the model's prediction (Sajjad, Kokhlikyan, Dalvi, and Durrani 2021). Hence, we can explain how the model makes its predictions with more human-interpretable concepts instead of only analyzing the influence of word tokens and low-level features on a prediction (Yeh, B. Kim, Arik, et al. 2020).

Three exciting works for this combined approach are TCAV (B. Kim, Wattenberg, Gilmer, et al. 2018) working with annotated concepts, ACE (Ghorbani, Wexler, J. Y. Zou, and B. Kim 2019) using k-means clustering to discover concepts, and ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020). The latter is also an unsupervised method being based on SHAP (Lundberg and S.-I. Lee 2017) not only extracting concepts but further analyzing *how much* a concept contributes to a prediction and guaranteeing a sufficient or *complete* explanation. We will adapt ConceptSHAP for our application and use it as the primary method of choice for our second set of explainability experiments. How it works in detail for our application is explained in section 4.2.2 for the methodology.

2.5. Contribution

We propose our approach, overcoming the shortcomings of state-of-the-art survey analysis, using methods from NLP and XAI, offering an extended analysis of the EMS 1.0 study. Our approach first predicts the variable from fine-grained (text) features and then explains the model predictions top-down either to a word and feature level or to a concept level, being more accurate and flexible.

First, we automate the process of quantifying text to make it useful for statistical analysis, heavily saving resources.

Further, our approach captures much more profound knowledge than traditional correlation analysis as we are using highly nonlinear models to extract relations and later explain them. With this, we can also very specifically identify the most critical text and numerical features in great detail.

While we want to examine characteristics in language from open-ended questions primarily, we also want to study correlations of other variables from the SCCT framework with the career goal variable. We, therefore, build a mixed prediction model using the latest deep natural language model BERT as introduced in section 2.3). It takes both qualitative and quantitative variables as input. This mixed prediction model allows us to analyze later open- and closed-ended survey answers while enabling shared insights from the correlation of both question types.

To generate a holistic explanation of the model and explore hidden patterns in students' language (beyond prominent coding schemes), we combine the latest explainability techniques based on SHAP introduced in section 2.4. First, we explain fine-grained relations on a feature importance level with SHAP values from Lundberg and S.-I. Lee (2017). With these, we study contextual word embedding and word token importance for open-ended questions and numerical feature importance for closed-ended questions.

Lastly, we test ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020), a new method, which combines concept-based methods (section 2.4.3) with element importance (section 2.4.4). With ConceptSHAP, we first want to extract latent concepts captured by certain neurons to measure the concepts' contribution to the model's career goal predictions. This directly relates to the traditional methods of manual open-coding and automated topic modeling but overcomes the limitations of these methods. The authors (Yeh, B. Kim, Arik, et al. 2020) from the method we use (ConceptSHAP) also claim a complete explanation, which states that all the information is captured by the concepts, independent of the number of topics.

This deep learning-based approach provides an improved process to standard study design and analysis, either eliminating the need for complex and hand-crafted constructs or evaluating them by extracting the most significant correlations between the

variables of interest automatically in hindsight. Further, our approach can efficiently generate more profound insights into open-ended answers, making it possible for studies to leverage qualitative data in combination with quantitative data more effectively than with open-coding or topic modeling methods for text analysis. Beyond offering an improved process for survey analysis, we further claim that it can be used for new survey design, as we can identify new concepts as influential factors for a variable of interest that can serve as inspiration for further surveys examining similar questions.

3. Survey Data

This work focuses on analyzing data coming from the longitudinal *Engineering Major Survey* (EMS) (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017). While the study in total consists of three surveys conducted between 2015 and 2019, we only focus on the data from the EMS 1.0 survey from 2015. 7197 students answered this survey in total. It contains answers from students enrolled at 27 universities across the United States. As described in 2.1, the whole study is based on the Social Cognitive Career Theory (SCCT) from Lent, Brown, and Hackett (1994). SCCT is a framework well established in educational research to study a student's career goal, the relevance of innovation in their future work, and how their decision-making is influenced by a total of 7 (in our case 8) topics such as educational experiences, background characteristics or self-efficacy (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017).

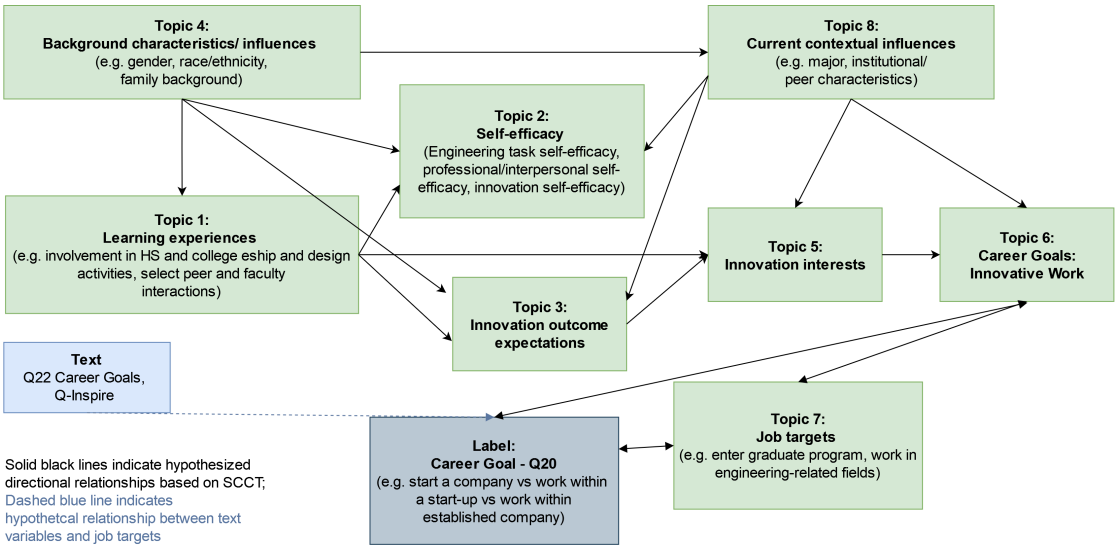


Figure 3.1.: The extended SCCT model adapted from the EMS 1.0 study (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017). Green nodes are the topics including the numerical feature variables Q22 and Q-Inspire, light blue node includes the text feature variables, dark blue node represents the label variable Q20

The SCCT framework proposes a functional relationship between those topics shown

in figure 3.1. Each of the topics is formed by one or several question variables, which can be either numerical - being our numerical features - or in the form of text - being our text features. The students' career goal - a multi-label categorical variable - will be our independent variable (label) to predict. Detailed information about the study research objectives, the methodology, and the respondent's characteristics can be found in the technical report from S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. (2017). In the following text variables, feature variables and labels will be described in more detail.

3.1. Independent Variables

The variable we are interested in for prediction is *Q20*:*"How likely is it that you will do each of the following in the first five years after you graduate?"*. It provides eight career possibilities, which will be our prediction heads:

1. Work as an employee for a small business or start-up company
2. Work as an employee for a medium- or large-size business
3. Work as an employee for a non-profit organization (excluding a school or college/university)
4. Work as an employee for the government, military, or public agency (excluding a school or college/university)
5. Work as a teacher or educational professional in a K-12 school
6. Work as a faculty member or educational professional in a college or university
7. Found or start your own for-profit organization
8. Found or start your own non-profit organization

Each of them can be answered with a Likert scale ranging from 0 *"Definitely will not"* to 4 *"Definitely will"*. We, therefore, retrieve eight different labels (L1 to L8) with five classes each, following a distribution shown in figure 3.2.

In order to simplify the classification problem, the five classes are binned for all of our experiments (see chapter 4) into two classes. Class 0 indicates low interest in the career goal of the respective prediction head, whereas class 1 indicates a high interest. To achieve a reasonable balancing of the labels, we tried two different strategies (as shown in fig. 3.3). We chose strategy two for binning the classes depending on the median of each label (see figure 3.3b). However, we can see that the classes are unbalanced in the multi-class case and the binned binary case. This can be a challenge for the model to

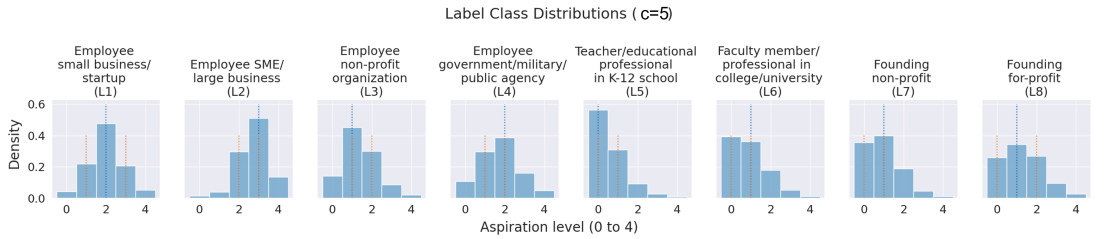
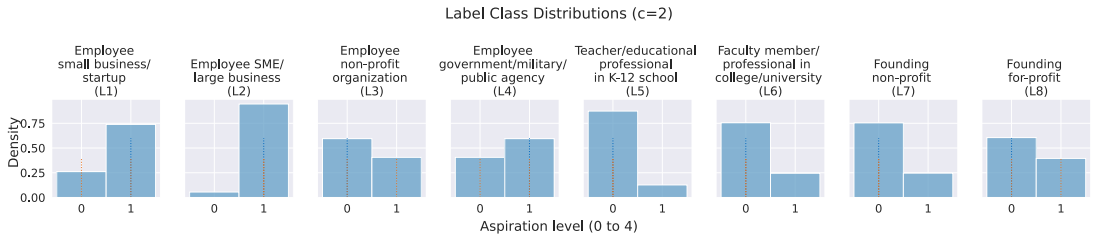
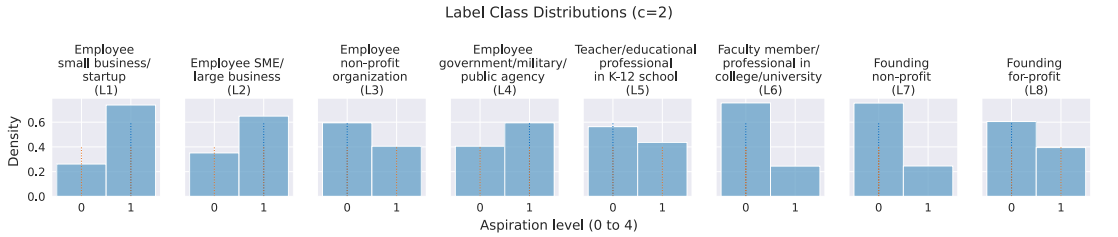


Figure 3.2.: The independent career goal variable Q20 has 8 different sub labels with a total of 5 classes (c=5) each. All eight labels show a clear bias towards lower labels.

learn to distinguish and predict the underrepresented class only from a few samples as described by Tantithamthavorn, Hassan, and Matsumoto (2020). To encounter this, we also conduct experiments as described in section 4.1, where we re-balance the classes during training.



(a) Split of (0,1) and (2,3,4)



(b) Split of (0,1) and (2,3,4) for median=0/1/2 and split of (0,1,2) and (3,4) for median 3/4

Figure 3.3.: Two strategies for binning the 5 classes into 2 classes (c=2), while aiming for a good balancing of the classes. Strategy 2 is the one chosen for the experiments of this work. We notice distribution improvements for label 2 and 5.

Figure 3.4 shows the linear Pearson correlation (Freedman, Pisani, and Purves 2007) between the labels. As this correlation is quite low (below 0.5 for most labels), this speaks for predicting each of the eight career labels as a unique task. We only take 6180 out of the entire set of 7198 samples for our analysis, which do not have a missing

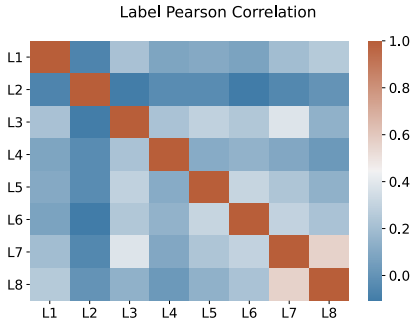


Figure 3.4.: Pearson Correlation between each of the 8 labels. The values range from 0.0 to 1.0, whereby 1.0 means a high positive correlation and 0.0 means no correlation. Neither of the labels shows a significant correlation (above a value of 0.5), suggesting to predict each of them separately.

or undefined value for any of the eight labels. This is also called list-wise deletion (Somasundaram and Nedunchezhian 2011).

3.2. Text feature variables

The two text variables we consider for this work are the following:

1. *Q22*: "We have asked a number of questions about your future plans. If you would like to elaborate on what you are planning to do, in the next five years or beyond, please do so here."

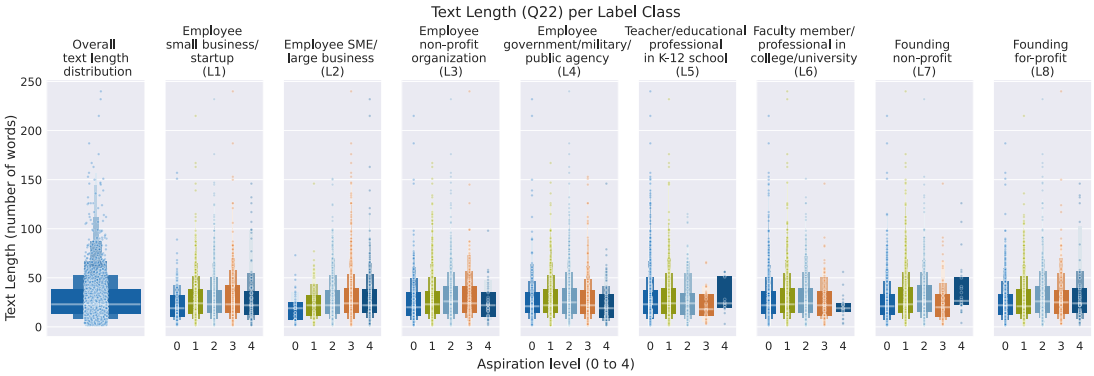
2. *Inspire*: "To what extent did this survey inspire you to think about your education in new or different ways? Please describe."

While *Q22* originally belongs to the job targets topic in the EMS 1.0 survey, both text variables are treated as an extra additional topic in this work (see 3.1), since we assume they provide extra information to the other topics, explaining a student's career choice. Opposite to the works of Levine, Björklund, S. Gilmartin, and Sheppard (2017) and Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken (2016) we are not performing qualitative research methods like manual open-coding (explained in section 2.2.1) on these text variables, but including them as quantitative data into our prediction model described in section 4.1.

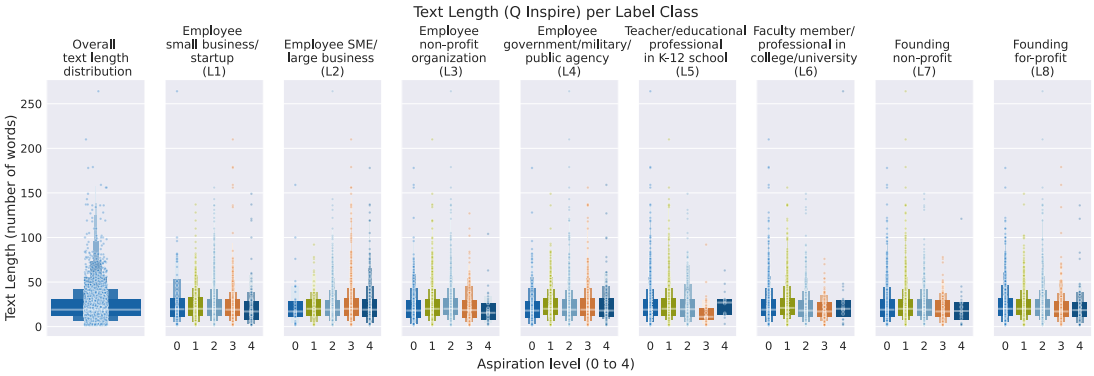
The text length statistics of *Q22* and *Inspire* are shown in figure 3.5.

In general, we cannot see any significant correlation of the text length with the label classes. As the performance of the model might be correlated to text length as shown

by Jafariakinabad, Tarnpradab, and Hua (2019), we can expect a similar performance of the text classification across all labels and classes.



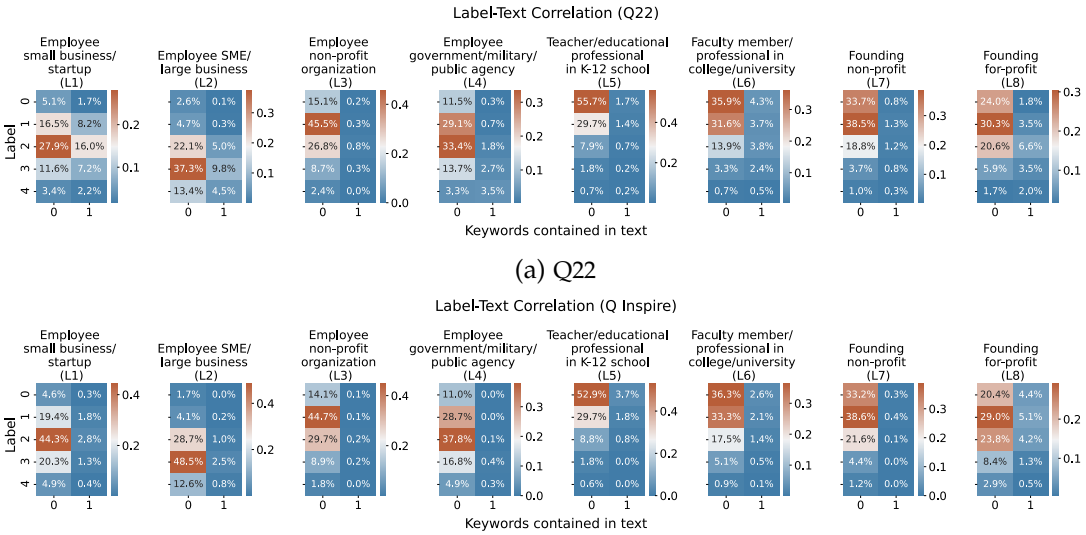
(a) Q22



(b) Inspire

Figure 3.5.: The figure shows the overall distribution of the text length (upper left), and the text length distributions grouped by the 5 classes for each of the 8 labels.

Before we try to predict labels from the text variables with a DNN, we are interested in whether we can extract a correlation between text and labels manually by searching for keywords matching a particular class in the text answers. For the text variable *Q22* we assume a strong keyword correlation since the answers are directly connected to the students' career choice. However, there are some correlations of keywords with lower classes as shown in figure 3.6. This means that students use those keywords, making them more likely to choose a higher class of 3 or 4. Nevertheless, they still show a low aspiration for this label (choosing class of 0, 1, or 2). For the second text variable *Inspire* on the other hand, we expect the correlation to be much more latent, which is why a keyword search does not indicate any correlation (see figure 3.6).



(a) Q22

(b) Inspire

Figure 3.6.: Correlation of key words related to each label with the classes. A strong key word to label correlation would show a strong diagonal pattern, which cannot be seen here.

3.3. Numerical feature variables

Besides text features from the survey's open-ended questions, we are also looking at a total of 119 numerical feature variables from categorical or five-point scale sub-question items composing 30 questions, which can again be attributed to the eight topics from figure 3.1 as shown in table A.1. The scale design (Likert scale), as well as the order of questions (e.g., asking background questions at the end of the survey), was chosen to minimize bias in survey response (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017).

As mentioned above, the numerical features belong to topic constructs from the SCCT framework shown in figure 3.1. According to this framework, the first topic *learning experiences* has only an indirect relationship to the *Career Goal: Innovative Work* (Lent, Brown, and Hackett 1994) and therefore also to our label, the job targets. This topic includes a broad set of experiences measuring opportunities for students first to experience personal performance achievements in certain areas, secondly to learn by observation and through social interaction with others and thirdly to perform relevant activities to gain positive or (negative) physiological experience (Lent, Brown, and Hackett 1994).

The second topic *self-efficacy* (see fig. 3.1) consists of two parts: *Innovation Self-Efficacy* (ISE) and *Engineering Task Self-Efficacy* (ETSE). Both variables are human-crafted constructs of multiple question items, defining a self-efficacy score from 1-4 (M. Schar,

S. Gilmartin, Harris, et al. 2017). The measurement is only as exact as its items. This can be limiting, especially in the case of innovation self-efficacy, which only focuses on innovative activities (M. Schar, S. Gilmartin, Rieken, et al. 2017). The items for the construct of ISE in the EMS 1.0 study have been adapted from Dyer, Gregersen, and Christensen (2019). The goal of this construct is to measure a student's confidence in performing activities that are related to innovation (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017) . The goal of the five items ETSE construct similarly is "to measure confidence in one's ability to perform integral technical engineering tasks" (M. Schar, S. Gilmartin, Rieken, et al. 2017, p. 5). *Innovation Outcome Expectations* (IOE), which is the third topic in fig 3.1, measures what students expect to be the outcome of innovative behavior (as defined by Dyer, Gregersen, and Christensen (2019)), in particular for "[asking] a lot of questions" (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017, p. 9).

The fourth, quite large topic *background characteristics/influences* includes, e.g., questions about family income, parents' educational degree, the role of entrepreneurship in a student's larger family or environment, a student's ethnicity or gender, or their type of citizenship.

Innovation interest, topic 5 in figure 3.1, being again a construct, extends what has been measured with the ISE and IOE constructs. While ISE and IOE focus on the two aspects idea generation or discovery of the innovation concept introduced by Kanter (2013) and Dyer, Gregersen, and Christensen (2019), *innovation interests* additionally measures the concepts from Kanter (2013) of "coalition-building", "idea realization", and "transfer or diffusion" (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017, p. 8).

The sixth topic *Career Goals: Innovative Work* (CGIW) is the primary outcome variable of the basic SCCT framework being the only topic in the SCCT framework directly connected to our label - the job targets. Hence, we expect a strong influence of this variable on the prediction, which we are going to explore in section 4.1. CGIW is again a construct composed of 6 items adapted from Kanter (2013) and S. G. Scott and Bruce (1994) from a more activity-based measurement to a goal-focused measurement. The goal of the CIGW topic is to measure a student's aspiration to choose an innovative career path (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017).

Topic 7, *current contextual influences* takes into account a student's current situation, for example, their grades, being an undergraduate or graduate, or their major and minor.

Works like the one from Atwood, S. Gilmartin, Harris, and Sheppard (2021) as mentioned in section 2.1 have already studied some of the correlations between, e.g., features like family income status, parents' educational degree, gender, ethnicity, and the student's college experiences (such as internships or study-abroad) or their engineering self-efficacy (ETSE). They point out the statistical under-representation

of URM students and, particularly, URM women in families with higher educational income or degree. Furthermore, they found a significant difference in engineering self-efficacy (ETSE) when looking at the intersection of gender and ethnicity (lower ETSE for URM women) or the intersection of family income and a college degree (lower ETSE for First-generation and low-income students)(Atwood, S. Gilmartin, Harris, and Sheppard 2021).

We are interested in whether the whole set of features and the labels correlate linearly or nonlinearly for our analysis. The stronger a linear correlation between features and labels, the simpler the prediction model could be. However, figure 3.7 shows a very weak linear correlation between those one-dimensional features (either binary or categorical) and the labels, indicating that the prediction of the Q20 career label is a complex task.

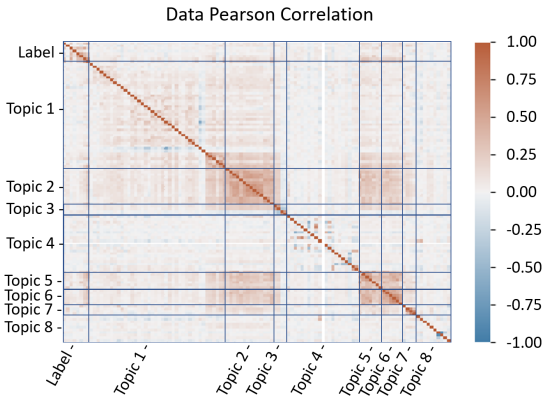


Figure 3.7.: Pearson Correlation between all of the one dimensional feature variables and the label: A value of 0 means no correlation while -1 and 1 would indicate a strong negative and positive correlation respectively.

4. Methodology

This chapter covers the two primary methodologies used in this work for answering the research questions introduced in chapter 1, focusing on the dataset introduced in chapter 3. Based on the two parts of the theoretical background in chapter 2.3 and 2.4, the methodology will also be split into two main parts of text classification and interpretation, answering the first and the second research question each as introduced in chapter 1.

4.1. Prediction of a Student's Career Goal from Open-Ended Answers

To answer our first research question of whether neural networks with a language model can extract information from a survey with open-ended and closed-ended survey questions to predict the primary variable of interest, we run several experiments with different model architectures. Since we are interested in how much the different variables and topics of the survey contribute to the prediction, our model architecture follows a modular, additive approach. This approach already introduces some weak form of explainability to the model, following the concept of perturbation-based methods introduced in section 2.4.2, in particular occlusion studies introduced by Zeiler and Fergus (2013) and J. Li, Monroe, and Jurafsky (2016).

We first build a pure text classification model for the text input described in section 3.2 (see section 4.1.1). As a second step, we build a classification model taking the numerical features described in section 3.3 as inputs (see section 4.1.2). The third step will be to combine these two model architectures to one large model architecture, which takes both text and numerical features as input (see section 4.1.1). For all three approaches, we test a regression head versus a classification head for the eight different label categories described in 3.1. For the regression head model, which simply has eight output neurons each predicting the direct labels' values, we use a mean absolute error loss (see eq. 4.1). For the classification head, on the other hand, which has eight heads with 2 to 5 class neurons, each to predict the class probability, we use a cross-entropy loss function (see equation 4.2).

$$MAE = \frac{1}{n}\sum_{i=1}^{N}\sum_{l=1}^{8}|\hat{y}_{l,i} - y_{l,i}| \tag{4.1}$$

$$CE = \frac{1}{n}\sum_{i=1}^{N}\sum_{l=1}^{8}\sum_{c=1}^{C} y_{l,c,i_{onehot}} \cdot log(softmax(\hat{y}_{l,c,i})) \tag{4.2}$$

The model is designed completely modular such that we can not only test different input and output variations, but also adjust its more detailed architecture like the number of hidden layers and units. The best set of architecture parameters is retrieved by an extensive tuning process. This set of parameters is used for all of the experiments section 4.1.1, 4.1.2 and 4.1.3.

Beyond these architecture modification experiments, we also performed some minor experiments. One of these experiments was to standardize the input (feature-wise z-score standardization, as suggested by Shanker, Hu, and Hung (1996)). Another one was to standardize the labels sample-wise (within-subject). The purpose of this experiment was to account for a person's individual bias for answering with a higher or lower score tendency (being common practice in psychology as discussed by Fischer (2004) and Greenleaf (1992)). The third experiment was to rebalance the dataset by down-sampling as all the eight labels show an unbalanced distribution for the two classes (refer to section 3, discussed by Tantithamthavorn, Hassan, and Matsumoto (2020)). While the feature standardization will be tested for the best (intermediate) models of all three sections 4.1.1, 4.1.2 and 4.1.3 as they have different inputs, the label standardization, and rebalancing will only be tested for the overall best model of section 4.1.3.

Another important topic is how the model deals with missing values in the dataset. Mislevy (1991) distinguishes between three different types of missing values: MCAR - values that are *Missing Completely At Random*, MAR - values for which the probability of being missing is related to another observed variable, and MNAR - values, which are *Missing Not At Random* meaning that their missingness is coupled to the variable for which they are missing. As MCAR occurs very rarely in the real world (Mislevy 1991), we assume MAR and MNAR to apply to our data.

We apply two different strategies to handle missing data. The first approach is list-wise deletion, where we delete any sample, which contains at least one missing value for any of the variables (Somasundaram and Nedunchezhian 2011). The second approach is constant imputation, where we insert constant values for the missing values and therefore keep the original size of the dataset (Somasundaram and Nedunchezhian 2011). We choose the value to be "-1" for the numerical features since many of our variables contain zero as a valid value, and a negative value gets ignored by the ReLU activation used for our network layers (see section 4.1.1).

As our dataset is already quite small with 7197 samples, we prefer the imputation method over list-wise deletion, where we lose a lot of samples and information, leading to poor performance and statistical accountability (Somasundaram and Nedunchezhian 2011). However, there are some cases where we perform list-wise deletion. The first case is for missing values in our label variables being a common practice in research (Somasundaram and Nedunchezhian 2011). This already reduces our dataset size to 6180 samples for any of the experiments. Secondly, we use list-wise deletion for the pure text feature models in section 4.1.1 and 4.1.3, since we have at most two variables for those models (Somasundaram and Nedunchezhian 2011). Imputing constant variables (an empty string for text data) when no or only one other variable contains additional information would keep "empty" samples without any predictive power. This method results in a dataset size of 1812 for the input of text variable *Q22*.

For the models, which combine text and feature input in section 4.1.3, we compare two different approaches: Performing constant imputation for all variables, including the text variables, versus performing imputation only for numerical features while still deleting the samples with missing values for the text variables.

4.1.1. Prediction from Text Variables

For the pure text classification, we use different approaches. We only focus on one of the two text input variables - *Q22* (described in section 3.2) - since the focus for the first experiment stage is to maximize the model performance by comparing different language model architectures described in section 2.3. The whole model architecture is shown in figure 4.1.

The main part of this architecture is the BERT language model (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019) introduced in section 2.3. As we only have a quite small dataset, we are not fine-tuning BERT (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019) but using a pre-trained version of the lightweight DistillBERT model (see section 2.3, Sanh, Debut, Chaumond, and Wolf (2019)) to create embeddings of the text instances, similar to Word2Vec (Mikolov, K. Chen, Corrado, and Dean 2013). We, therefore, only train the layers after the BERT model, which are simple, fully connected layers with a ReLU activation. While the main layers of the BERT model are used without any adaption, a central part of this experiment is to try different variations of extracting and processing the contextual word and sentence embeddings.

We tried three main approaches as indicated in figure 4.1 with the *XOR* option. For one student answer (called a sentence, but can be arbitrary continuous text), which is a sequence of word tokens ($\in \mathbb{R}^{250}$), the BERT model outputs 250-dimensional hidden state vectors $\in \mathbb{R}^{250\times768}$). The first token, called "CLS", is a special token aggregating the whole sequence (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019). Therefore,

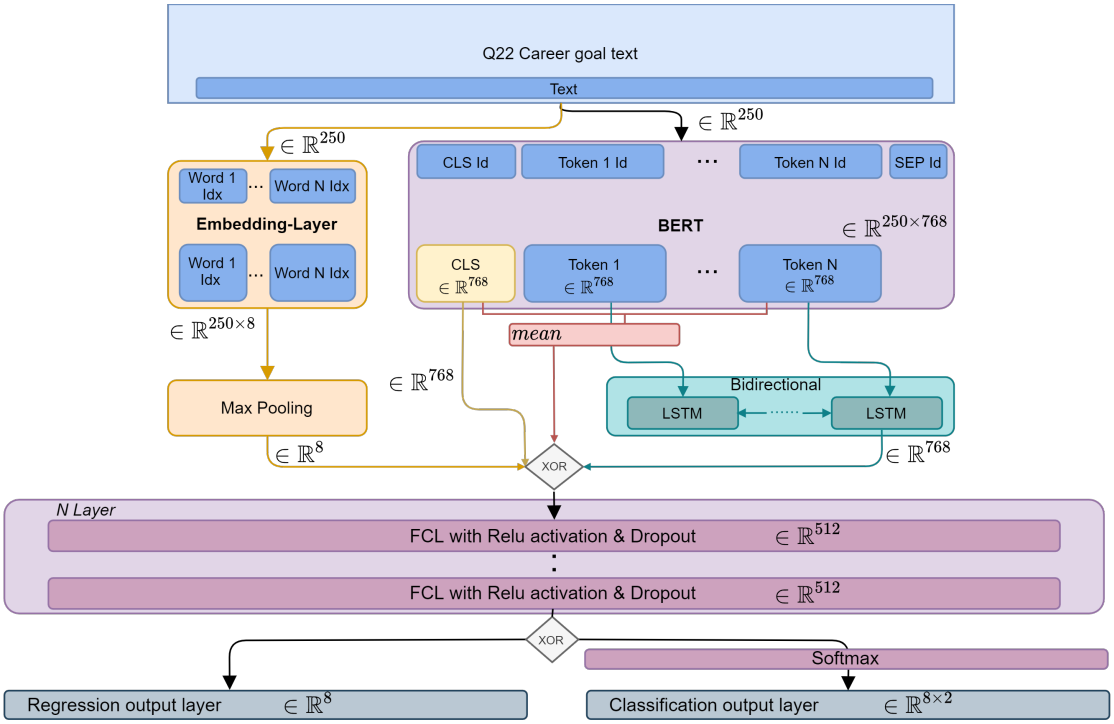


Figure 4.1.: Model architecture for pure text classification. The XOR part signals the different model architecture choices

the first and easiest approach is to take the embedding vector ($\in \mathbb{R}^{768}$) encoding the "CLS" token as further one-dimesnional input to the fully connected layers. This is common practice in literature and has also been proposed by the authors of the BERT model themselves (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019). The second approach for retreiving a one-dimensional embedding vector encoding the overall sentence ($\in \mathbb{R}^{250\times768}$) is to average over all of the 250 word token embedding vectors, which has also been suggested by Wolf, Debut, Sanh, et al. (2020). The third approach is slightly more complex as it introduces another layer to the network. For this approach, each of the 250 word token embedding vectors ($\in \mathbb{R}^{250\times768}$) is fed individually into a BiLSTM layer (introduced in section 2.3, (Graves and Schmidhuber 2005)), which then again outputs a single one-dimensional vector of a size that can be chosen manually. In our case, it has been set to a vector of size $\in \mathbb{R}^{512}$ for all the experiments. This approach tested whether it makes a difference to process the information from each word token embedding vector individually on a more detailed level or take the overall sequence information on a more abstract level as for the first two approaches. However, the BiLSTM layer adds a higher capacity to the model, making it prone to overfitting.

Besides those three experiments based on the pre-trained BERT model, we further tried a completely different model, where the BERT model was substituted with a single embedding layer (as described in section 2.3) that was trained from scratch on the dataset text corpus (see fig. 4.1). The goal of this embedding layer is to convert input word tokens in the form of integers into continuous embedding vectors ($\in \mathbb{R}^{8}$). The layer is similar to GloVe (Pennington, Socher, and Manning n.d.) or Word2Vec (Mikolov, K. Chen, Corrado, and Dean 2013), while those methods deliver already ready-to-use embeddings. Opposed to the BERT model, the word inputs for this layer were not encoded with the pre-trained BERT tokenizer (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019), but simply as an integer of a vocabulary set of size 1000 based on the text input.

Overall, the pure text classification part entails four experiments - three for the different embedding post-processing methods "CLS", "mean" and "BiLSTM" and one for the "embedding layer". As we have two different heads, we run those four experiments for both of them, which results in eight experiments in total. The results for the experiments of this section are presented in section 5.1.1.

4.1.2. Prediction from Numerical Feature Variables

After testing several model architectures for the text part, the second part tests a model, which only takes the numerical features as inputs (see fig. 4.2).

As shown in figure 4.2, the different numerical feature variables are grouped by the topic they belong to, which can be fed to the model in a modular way. One way

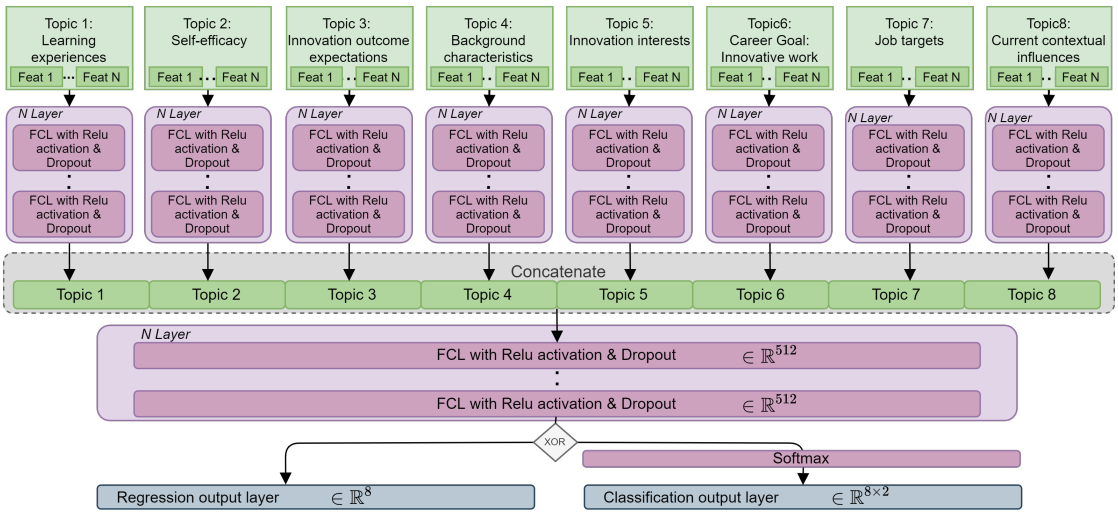


Figure 4.2.: Model architecture for pure numerical feature classification. The number of fully connected layers in the different topic streams is completely modular and can also be 0, which results in direct concatenation of the input topics.

of processing the different topics is to feed them through separate model streams consisting of a flexible number of fully connected layers and to concatenate them before feeding them again through a final stack of fully connected layers. This is similar to how we process the different text variables in the text model seperately before concatenating them at the embedding layer (compare figure 4.1 and figure 4.2).

The other way is to concatenate all of the topics directly (without any layer in the separate streams shown in fig. 4.2) and process them all together, which is equivalent to directly joining all the information and dissolving the topic structure. This already helps us to understand, whether the model makes use of the imposed topic structure by the SCCT (Lent, Brown, and Hackett 1994) framework.

So the different experiments of this section will be to first test each topic individually and then, in a second step, to test all of them together in a combined fashion. For this second step, we then run one experiment with the topics processed in different streams and one experiment, with the input topics concatenated directly before the first layer.

While most of the features are ordinal or binary and can therefore be fed as one value to the model, other variables have to be one-hot encoded as they are nominal without any meaningful relationship between the categories of this variable. Even though the ordinal and binary values already inherently carry some ordered relationship information, we can even one-hot encode them too. This enables us to try whether the neural network extracts a different relationship between the numerical feature values. We, therefore, also run this small experiment for the best model from the major

experiments before.

4.1.3. Combined Prediction from Text and Numerical Feature Variables

While the first two experiment sections focus on which model architecture works best for either pure text feature or pure numerical feature classification, we want to combine both models in this section. The text and numerical feature streams are joined in one model as shown in figure 4.3.

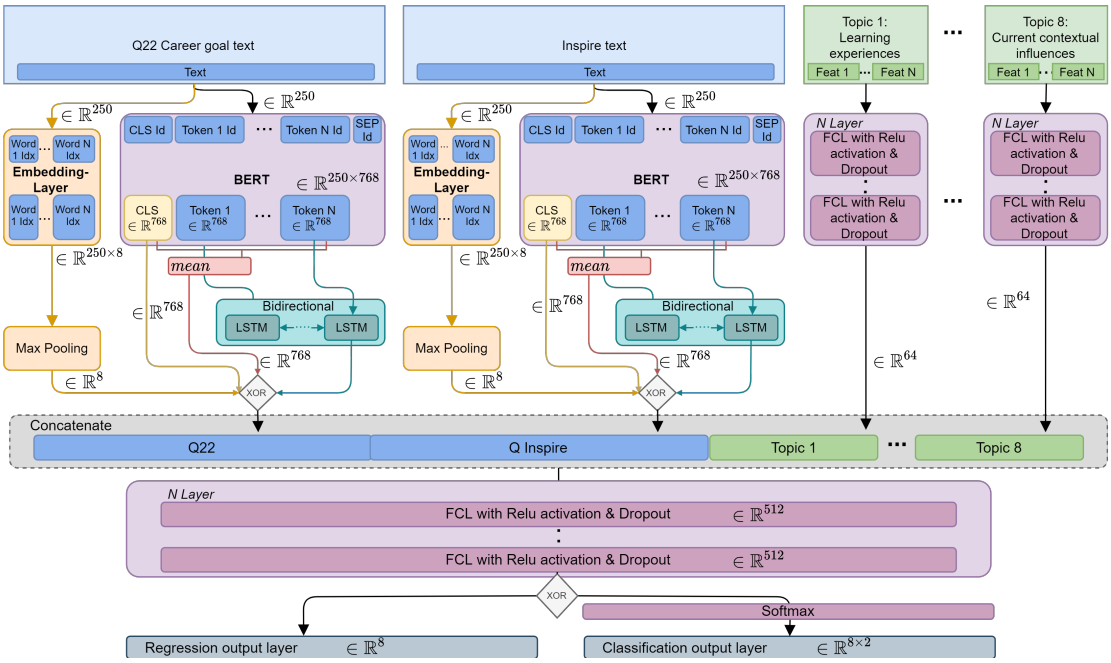


Figure 4.3.: Model architecture combining bot text and numerical feature classification architectures. The XOR part signals again the different model architecture choices, while however for this experiment the architecture itself is fixed (taking the best results from section 4.1.1 and 4.1.2 and only the input itself is varied.)

While this model is again modular as the models above, in this experiments section, we only vary the various input possibilities while fixing the best architecture version of both the text and the numerical feature model for the overall architecture. We, therefore, compare the following input variants:

Only text variable *Q22*, the *inspire* text variable, both text variables combined, only the best topics combination, the best topics combination combined with the text variable *Q22*, the best topics combination combined with the text variable *inspire* and lastly the best topics combination together with both text variables *Q22* and *inspire*. For

the best version of this experiment, we are then testing label rebalancing and label standardization across samples to account for individual bias as described in the introduction for the text classification experiment, section 4.1. For the best of those models, we additionally tune the set of hyperparameters to get the best overall result as explained in section 5.1.3.

4.2. Interpreting the Predictions

In order to understand the model's prediction holistically, we apply several methods explaining the model and its input on different levels. The first set of explanations will focus on local and global causation analysis based on SHAP, introduced in section 2.4.2. We want to highlight the critical input features and the inner neurons belonging to the text embeddings.

The second part will combine concept-based analysis with causation analysis to get a higher-level understanding of how the model captures information and uses it to make predictions. For this reason, we use the ConceptSHAP method as introduced in section 2.4.3 to first extract concepts captured by the model. We then use the idea of SHAP to explain their global importance on the different prediction heads.

4.2.1. Low-level Feature and Neuron Explanations with SHAP

Using SHAP, we explain the local and global influence of different parts of the model on the model's prediction, as shown in figure 4.4. For simplicity, we will only present explanations for the eighth prediction head. It is the head with the best performance and captures the likelihood of starting a "for-profit company", which we are most interested in.

For the first set of experiments, we build on the modular architecture approach from section 4.1, which allows us to already identify the most important input on a general numerical feature topic and text variable level based on the model's accuracy. However, to get to a more profound explanation level at this step, we identify which input information dominates the model's prediction on a detailed feature level.

We, therefore, split the model at the text embedding level and treat the embeddings as the model's input together with the numerical features. As a first experiment, we then calculate and compare the SHAP values for these features - numerical or embedding ones - globally and for local predictions. The second experiment looks at the second part of the model from the embedding layer to the text input. It reveals which parts of the text input trigger the neurons with the highest activation from the first experiment on a local prediction level. The third experiment is again only locally and examines the text input level but looks at the SHAP values directly for the predictions. We run

the second and third experiments to get a sense of how the influence of the text tokens differs when we look at the whole model instead of opening it on an embedding level and looking at this internal level first.

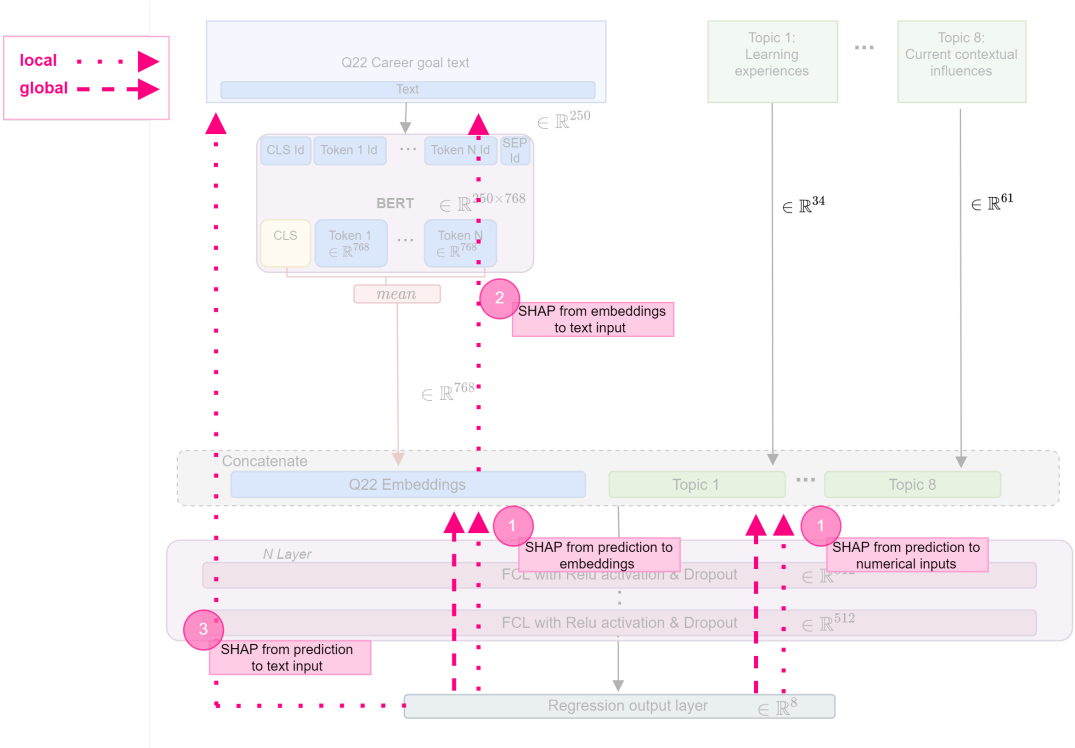


Figure 4.4.: Explainability experiments with SHAP values for different parts of the model. (1) Global and local SHAP values from prediction to intermediate layer with embeddings and numerical features as inputs, (2) local SHAP values from embeddings to text input, (3) local SHAP values from prediction to text input

4.2.2. Higher-level Concept Explanations with ConceptSHAP

For the second set of explanation experiments, we want to understand how the model captures higher-level information from the text input and uses this condensed information to make predictions. For this purpose we use ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020) introduced in section 2.4.3. This approach reveals the higher-level information captured by the model in the form of concepts. These concepts are defined by concept vectors c_j, which have a high similarity measure to a set of word embeddings forming a concept as the nearest neighbors of this vector. Therefore, the number of concepts is equal to the set of concept vectors ($c = \sum_j^N c_j$) and can be defined manually. While the

number of concept vectors is pre-selected, they are trained in an unsupervised fashion sharing some similarities with clustering approaches.

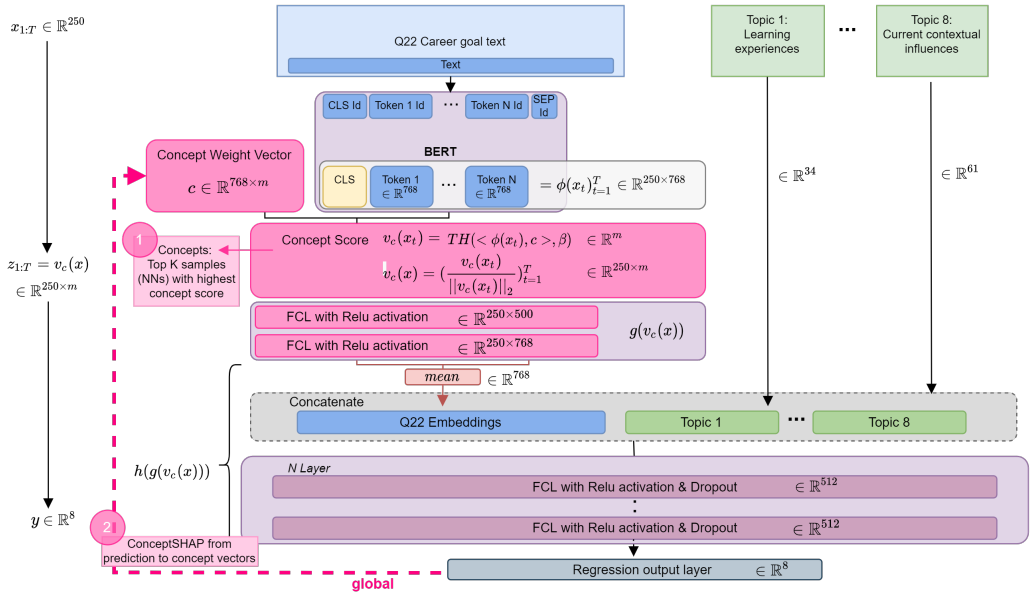


Figure 4.5.: Explainability experiments with a concept-based method called ConceptSHAP. The original model is extended to a surrogate model to train concept vectors c_j, which function as the centroids of the concepts. These concepts are then being formed by the top k nearest neighbour tokens embeddings to the concept vectors (1). In addition to the pure concept extraction, we can measure their importance for the prediction of the model by using the principle of SHAP, (2).

When adapting ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020) to our model, the method works as following: We split the original model $f(x)$ into two parts at the word embedding layer (as shown in figure 4.5) such that $f(x) = h(\phi(x)$ with $\phi(\cdot)$ being the first part from input up to the embedding layer and $h(\cdot)$ being the upper part from embeddings to the prediction. While for the SHAP experiments we split the model at the word embedding layer after having computed the mean embedding vector ($\in \mathbb{R}^{768}$), for ConceptSHAP we split it before taking the mean vector getting embedding vectors $\phi(x_t)$ for each word token (($\in \mathbb{R}^{250\times768}$)). We then introduce some intermediate layers between the two parts (as shown in figure 4.5) with the concept vectors $c = \sum_j^M c_j \in \mathbb{R}^{768\cdot m}$ being treated as a trainable weight layer. Computing the inner product $< \phi(x_t), c_j >$ gives us a measure of whether the token embedding is close to concept $c_j \in \mathbb{R}^{768}$, also defined by the authors as the concept score of a token embedding. We additionally define a minimum threshold value β for the dot products, which lets us take only sufficiently close token embeddings into account. The concept score for one token embedding

can therefore be computed as $v_c(x_t) = TH(< \phi(x_t), c_j > \beta)_{j=1}^{M} \in \mathbb{R}^m$. Applying some normalization for numerical stability leads to the following overall concept score for all embedding tokens $v_c(x) = (\frac{v_c(x_t)}{||v_c(x_t)||_2})_{t=1}^{T} \in \mathbb{R}^{T\cdot}$.

The goal is to find the underlying concepts for which we can recover the prediction from the original model $f(x_t)$ by projecting the concept scores $v_c(x)$ back to the activation space $\phi(x)$ with a mapping function g: $\mathbb{R}^{T\cdot m} \longrightarrow \mathbb{R}^{T\cdot 768s}$ such that $f(x_t) \approx h(g(v_c(x_t))$. The success of recovering the original prediction with this surrogate model using an identified set of concept vectors $c_1, ..., c_m$ can also be seen as their completeness of capturing the high-level information, being defined as:

$$\eta_f(c_1, ..., c_m) = \frac{sup_g \mathbb{P}_{x,y\ V}[y = argmax_{y'} h_{y'} g(v_c(x))] - a_r}{\mathbb{P}_{x,y\ V}[y = argmax_{y'} f_{y'}(x)] - a_r} \tag{4.3}$$

with $sup_g P_{x,y\ V}[y = argmax_{y'} h_{y'} g(v_c(x))]$ being the best accuracy of the model with a certain set of concept vectors $c_1, ..., c_m$ and a_r being the random accuracy, which is $a_r = 0.5$ in our case of two classes. As stated by Yeh, B. Kim, Arik, et al. (2020) this introduction of concept completeness also differentiates ConceptSHAP from other concept based methods like TCAV (B. Kim, Wattenberg, Gilmer, et al. 2018) or ACE (Ghorbani, Wexler, J. Y. Zou, and B. Kim 2019) introduced in section 2.4.3.

The objective function for identifying the concept vectors $c_{1:m}$, therefore, aims at maximizing the completeness of the discovered concepts with a surrogate loss optimizing for both $c_{1:m}$ and the layers of the mapping function g:

$$\log_{c_{1:m}, g} \mathbb{P}[h_y(g(v_c(x)))] \tag{4.4}$$

Unlike other clustering methods like k-means, we want to ensure further that the concepts are interpretable and semantically meaningful. For this, the authors of ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020) propose to add an additional regularizing term $R(c)$ to the objective function in order to encourage the spatial dependency of concepts:

$$R(c) = \lambda_1 \frac{\sum_{k=1}^{m} \sum_{x_a^b \subseteq T_{c_k}} \phi(x_a^b) \cdot c_k}{mK} - \lambda_2 \frac{\sum_{j \neq k} c_j \cdot c_k}{m(m-1)} \tag{4.5}$$

With $x_a^b \subseteq T_{c_k}$ being the top K nearest neighbors of token embeddings to concept c_k, the first part ensures spatial closeness to the concept, while the second part ensures the different concepts to be spatially as far as possible.

This method of optimizing a surrogate model which first outputs a score $v_c(x_t)$ for each text token x_t belonging to a specific concept m and then making the prediction based on this concept score is equivalent to the principle of topic models with the concept assignment $x_t \longrightarrow z_t$ and the prediction model $z_t \longrightarrow y$.

Now that we have the concept vectors, we can compute the SHAP values for these concepts similar to the original concept of SHAP as defined in section 2.4.2 (equation 2.9). However, instead of perturbing the text input, we treat the concepts as the input to the "topic model", measuring the contribution of a certain concept to the model as its influence on the completeness score $\eta(c_i)$. Given a set of concept vectors $C_S = c_1, c_2, ..., c_m$ we compute the ConceptSHAP values as the following:

$$s_i(\eta) = \sum_{S \subseteq C_S c_i} \frac{(m - |S| - 1)!|S|!}{m!} [\eta(S \cup c_i) - \eta(S)] \tag{4.6}$$

Like this, we can measure the global influence of the concepts on each prediction head. Having the concept with its top k nearest neighbors, we not only investigate the concepts' structure and their global influence on the prediction. We also run an experiment for whether the nearest neighbor samples belonging to a concept show similar embedding activation patterns for their "mean" embedding. This indicates that the concepts capture semantically coherent information.

5. Results

In the following, the results for the experiments described in the methodology chapter 4 are presented. We follow the same structure as for the methodology, first presenting the results for predicting career goals from the students' answers with a deep NLP model and second explaining these predictions with methods from the field of XAI.

5.1. Prediction of a Student's Career Goal from Open-Ended Answers

The goal of this section is to analyze the results for the experiments described in 4.1. We want to evaluate and compare the model architectures' performance to predict the eight career labels from Q20. The best model will then be further analyzed with explainability methods in section 5.2.

The metric of choice to measure the prediction performance is the macro f1 score for both the classification and the regression head. The macro f1 score is computed as the mean of the individual class f1 scores (as favored as one of two options to compute the macro f1 score by Opitz and Burst (2019)), which is again the harmonic mean from precision and recall of the individual classes. The formula is the following:

$$f1_{macro} = \sum_{c}^{Classes} 2 \cdot \frac{precision_c \cdot recall_c}{precision_c + recall_c} \tag{5.1}$$

The macro f1 score is particularly useful for unbalanced classes (see our dataset as shown in figure 3.3) as it takes into account the f1 score and assigns the same importance to both classes, even though one of the classes is underrepresented (Opitz and Burst 2019). We, therefore, even get a bad f1 score when the model performs good on the predominant class but bad on the minority class.

For the classification head, precision and recall and therefore the class f1 can be directly computed from the confusion matrix inherently coupled with the class predictions. For the regression head, we first have to round the predictions to an integer to get the confusion matrix, from which we can again compute the three metrics.

5.1.1. Prediction from Text Variables

As described in section 4.1.1, we conducted eight main experiments for the pure text classification. The resulting macro f1 scores for the eight class labels of the test data are presented in table 5.1. The results are presented for the four different model architectures *CLS*, *mean*, *BiLSTM*, *embedding* with a classification (C) and regression (R) head each. We distinguish the results for the eight different prediction labels (L1 to L8). As we can see, the macro f1 score is only 63.70% in the best case for label 8, which represents the class "founding your own for-profit company". This result already indicates that the correlation between the text answers and the labels is rather weak. As the best results are achieved for label 8, we will focus on this label for our comparison of the models.

Among the four different architecture choices, the *mean* embedding model performs best. One explanation for the *mean* model performing better than the *CLS* model might be that with the *mean* we take the information of all token output vectors into account, while for the *CLS* model, we only take the information of that one vector. The *BiLSTM* layer on top of the BERT model (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019) and the *embedding* layer without BERT both add additional parameters that must be trained. Since the *BiLSTM* model performs well on the training data, the bigger model capacity could be a possible explanation for a worse generalization on the test data. We cannot state a clear tendency of whether the classification head performs better or the regression head.

Additionally, we performed the experiments of embedding standardization as described in section 4.1.1 for the best model and label 8: The model with feature standardization results in a macro f1 score for label 8 of 56.96% for the regression head and 58.61% for the classification head. These results are worse than the macro f1 scores for the model without any of those measures (63.70% (C) and 62.40% (R)).

Given the results, we choose the architecture with the *mean* of the BERT embeddings (Devlin, M.-W. Chang, K. Lee, and Toutanova 2019) without feature embeddings standardization as our best model for further experiments in section 5.1.3.

5.1.2. Prediction from Numerical Feature Variables

The macro f1 scores for the classification of the numerical features are presented in table 5.2. As described in section 4.1.2, we tested all of the topics as individual inputs before testing them in a combined fashion with separated streams (sep.) or directly concatenated (dir.). The results are presented for these experiments with a classification (C) and regression (R) head each. We distinguish the results for the eight different prediction labels (L1 to L8).

		L1	L2	L3	L4	L5	L6	L7	L8
CLS	C	57.12	58.05	48.49	48.17	42.26	46.42	44.74	60.66
	R	54.05	51.26	36.41	44.24	35.21	42.74	43.44	53.96
mean	C	51.66	**60.10**	**56.89**	44.61	**48.40**	**51.85**	**52.50**	**63.70**
	R	53.82	51.36	50.82	**58.75**	43.63	42.24	46.71	62.40
BiLSTM	C	42.75	38.74	39.17	37.73	35.36	43.11	42.18	37.88
	R	52.82	54.49	36.70	49.77	34.91	42.38	42.62	58.18
embedding	C	**54.57**	47.62	50.52	50.06	48.31	48.05	46.45	49.66
	R	52.21	47.68	47.83	43.04	48.06	43.56	51.22	50.27

Table 5.1.: Macro f1 score in % for the different architectures of the pure text model (variable *Q22*) with classification (C) and regression (R) head each. Results are presented for all of the 8 labels.

Table 5.2 shows that the best results are obtained for the model where all topics features are combined directly. Similar to the results for the text classification model of section 5.1.1, we can see that label 8 - "Founding for-profit" - performs best among all labels with an f1 score of 74.65% (R). Comparing classification versus regression does not give a clear result as for the text classification, but we obtained the best result with the regression head.

Comparing the f1 scores of label 8 for the individual topics, we see that topic 1, "Learning experiences", topic 5, "Innovation interests" and topic 6, "Career goal: Innovative work" perform best. We, therefore, expect those topics to have the highest importance for the model including all topics. In order to test this assumption, we again train a model that only includes topic 1, topic five, and topic 6. We want to see whether we get a similar or better result than with all the topics as inputs. The f1 score for the eight prediction head of this model is 70.46% (R) and 72.05% (C), which is quite close to the result with all topics of 74.65% (R) and 72.65% (C). This observation reinforces our hypothesis that only topics 1, 5, and 6 contribute to a large extent to the prediction. However, the model with all topics still performs slightly better, which is why we keep it as our best model architecture for further experiments. The fact that topics 1, 5, and 6 have the highest contribution also aligns to some extent with the adapted SCCT model (Lent, Brown, and Hackett 1994) from the EMS 1.0 survey (S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. 2017). It suggests a direct connection between topic 6, "Career goal: Innovative work" and our independent variable, "Job targets". The "Learning experiences" of topic one and the "Innovation interests" of topic 5 are, however, not directly coupled with the job targets in this framework from

Lent, Brown, and Hackett (1994) and adapted from S. K. Gilmartin, H. L. Chen, M. F. Schar, et al. (2017). Therefore, the findings of our model bring in a new perspective. However, the model does not process features belonging to a topic in different streams but combines all features on the input level, indicating that the model learns better data representation without the topic structure induced by the SCCT framework. The detailed contribution will be discussed in section 5.2.1.

Running again the tests of feature standardization for the best model with all topics combined for label 8 gives an f1 score of 66.45% and 71.79%, for classification and regression, respectively, compared to 72.65% and 74.65% for classification and regression without standardization.

As explained in section 4.1.2, we have only one-hot encoded some of the features but also tested the case of one-hot encoding all of the numerical features. This gives an f1 score for the model with all topics of 67.56% (R) and 68.77 (C) for label 8. It is worse than our standard encoding, where most features are kept ordinal.

As any of the measures give a better result, we choose the model without complete feature one-hot encoding and feature standardization that takes all topics as numerical inputs as the fixed architecture for the next experiment of section 5.1.3.

5.1.3. Combined Prediction from Text and Numerical Feature Variables

For the final experiments, we compare all possible input combinations presented in table 5.3: The best results for the pure text classification of variable *Q22* (Q22, no T) and the pure numerical feature topic classification (No text, all T) versus the different model combinations of the text variables *Q22* and *Inspire* (Ins.) and numerical feature topic inputs (all T, no T). The results are presented for all of these experiments in table 5.3 with a classification (C) and regression (R) head each. We compare the results for the eight different prediction labels (L1 to L8).

Table 5.3 shows that the best results are again obtained for label 8 (as already stated in section 5.1.1 and 5.1.2). For the f1 scores of this label, the model which combines *Q22* and the feature topics achieves the best result of 75.02% (C)/ 75.03% (R). However, this result is only slightly better than the pure numerical feature model's - a macro f1 score of 74.65% (R). This observation indicates that the text variable *Q22* only contributes slightly positively to the model's prediction performance. The text variable *Inspire* even contributes negatively as it deteriorates the f1 score for the model with both text variables and the topics to 73.61% (R) (in the best case with the regression head). The first six results for the pure text classification models in table 5.3 even reinforce this observation, as all of them perform worse than the pure numerical feature topic model. The architecture with text variable *Inspire* only shows the worst results for the text models with 42.69% (C) and 35.48% (R), while the model with text variable *Q22* shows

			L1	L2	L3	L4	L5	L6	L7	L8
topic 1		C	44.39	41.74	57.46	41.36	52.21	49.62	58.07	66.83
		R	41.63	40.48	44.78	42.77	44.04	44.92	46.51	64.92
topic 2		C	48.42	44.83	54.36	42.51	40.32	42.60	42.74	62.39
		R	43.56	39.98	36.46	38.16	35.17	43.68	43.09	55.41
topic 3		C	42.74	46.80	48.03	42.17	54.85	46.05	48.10	50.18
		R	43.33	39.68	45.84	38.02	48.54	46.42	47.09	48.60
topic 4		C	42.28	39.33	45.18	47.16	45.22	44.94	44.71	54.37
		R	42.17	40.39	44.03	51.85	41.54	48.91	48.24	48.06
topic 5		C	44.94	51.00	58.68	45.98	55.64	46.77	43.44	64.33
		R	44.33	48.75	53.58	41.33	51.86	43.06	43.51	62.98
topic 6		C	49.26	40.35	44.68	47.18	38.39	42.71	42.57	57.70
		R	44.36	44.39	37.53	38.89	35.85	42.46	43.16	61.96
topic 7		C	46.40	61.60	56.66	46.20	54.11	58.03	43.02	44.29
		R	47.32	**62.69**	50.31	51.28	52.65	60.86	43.59	48.98
topic 8		C	46.41	44.39	52.06	51.84	45.58	45.68	44.04	48.69
		R	48.92	56.72	49.96	53.97	49.65	53.29	43.37	38.91
all topics	sep.	C	51.41	60.80	60.90	**57.35**	**61.06**	60.79	59.29	70.25
		R	**51.81**	55.66	52.38	56.31	52.84	55.83	53.32	67.74
	dir.	C	50.85	53.34	61.03	52.40	57.03	**67.88**	**61.02**	72.65
		R	50.79	54.17	**61.58**	57.33	58.94	56.92	59.08	**74.65**

Table 5.2.: Macro f1 score in % for the different model architectures of numerical feature topic inputs with classification (C) and regression (R) head each. Results are presented for all of the 8 labels.

the best results for the text models with 63.70% (C) and 62.40% (R). The performance of the combined version with both text variables lies in between, scoring 51.12% (C) and 58.73% (R).

The dataset size for the experiments in table 5.3 has varied depending on the input as explained in the introduction of section 4.1. The experiments for the combined models shown in table 5.3 are all performed with only one of the missing data strategies introduced in section 4.1: Imputing constant values for the numerical features and performing list-wise deletion for the text variables. We, therefore, want to compare the results for the best model following the strategy described above with the strategy of using imputation with constant values (being an empty string in the case of a text variable) for all of the variables, losing none of the data samples. The respective macro f1 scores are 72.53% (C) and 71.62% (R). This result is slightly worse than the version without constant value imputation for the text variables, which is why we choose the latter one. Nevertheless, one must say that the difference is relatively small since the text variable *Q22* does not contribute much to the result in any case.

The best model architecture is the one using all numerical feature topics and *Q22* as input variables, BERT for extracting the *mean* sentence embedding vector, and a regression output head for prediction. We perform the label rebalancing and within-sample label standardization experiments for this model. These result in macro f1 scores of 70.66% (C)/69.22% (R) and 64.14% (C)/63.37% (R) for label 8 respectively. This result is worse than without any of the measures. Our overall best result delivers the model without label rebalancing and within-sample label standardization.

We further tuned this model by testing an extensive set of the detailed model architecture and training parameters. The detailed model architecture looks the following: Four hidden layers after concatenating contextual text embeddings and numerical features, no hidden layers for the numerical feature streams meaning direct concatenation with text embeddings, 250 hidden units per layer, and no batch normalization but dropout at every layer with a probability of 0.05. We use a batch size of two for training the model, a learning rate of 0.0004, and no regularization.

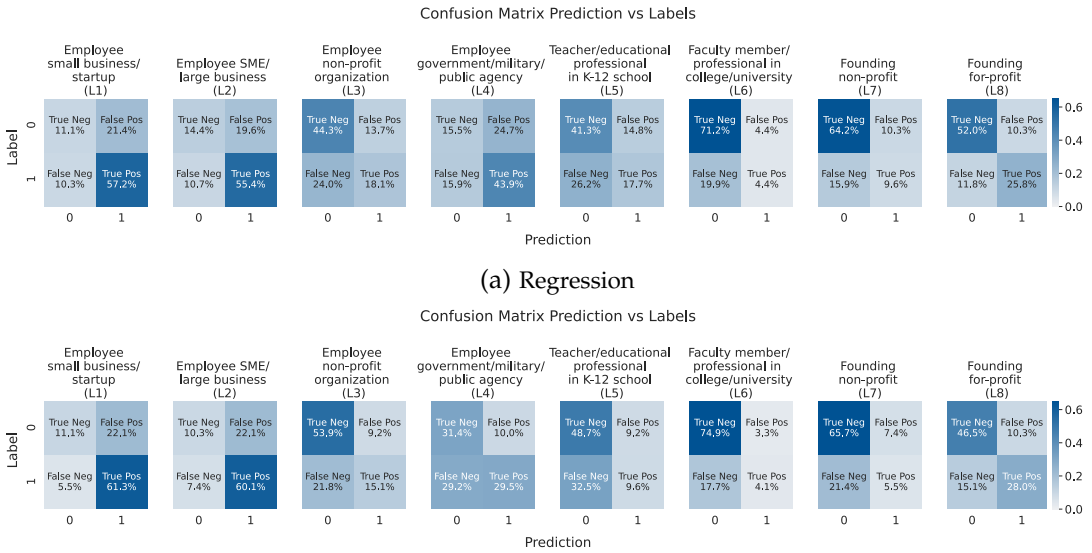
Overall, the resulting f1 performance score remains quite low, as already noted in section 5.1.1 - especially since we could only achieve a good result for the eighth prediction head. While our model can learn well on training data, it does not perform well on the test data.

5.2. Interpreting the Predictions

In the following, we only present explanations for the best-performing model from the section above. Furthermore, we focus on explanations for the eighth prediction head

			L1	L2	L3	L4	L5	L6	L7	L8
Q22	no T	C	51.66	60.10	56.89	44.61	48.40	51.85	52.50	63.70
		R	53.82	51.36	50.82	58.75	43.63	42.24	46.71	62.40
Ins.	no T	C	46.66	38.20	40.68	42.20	50.21	43.48	46.08	42.69
		R	42.26	39.79	36.07	37.77	37.10	41.79	41.88	35.48
Q22+Ins.	no T	C	45.69	59.87	52.31	53.11	47.92	59.71	50.91	51.12
		R	**63.48**	47.46	50.59	45.20	41.06	41.29	39.86	58.73
No text	all T	C	50.85	53.34	61.03	52.40	57.03	**67.88**	61.02	72.65
		R	50.79	54.17	61.58	57.33	**58.94**	56.92	59.08	74.65
Q22	all T	C	63.01	60.74	**63.53**	**60.87**	50.77	57.76	54.90	73.64
		R	59.69	**63**.64	59.59	55.84	56.62	56.03	**62.66**	**76.23**
Ins.	all T	C	57.23	59.08	57.63	54.22	54.68	57.48	65.30	69.24
		R	48.33	47.00	51.49	50.45	48.92	46.12	58.49	72.47
Q22+Ins.	all T	C	58.71	57.52	59.86	55.51	55.16	58.56	62.40	71.55
		R	59.49	54.62	63.27	55.50	56.83	49.58	56.60	73.61

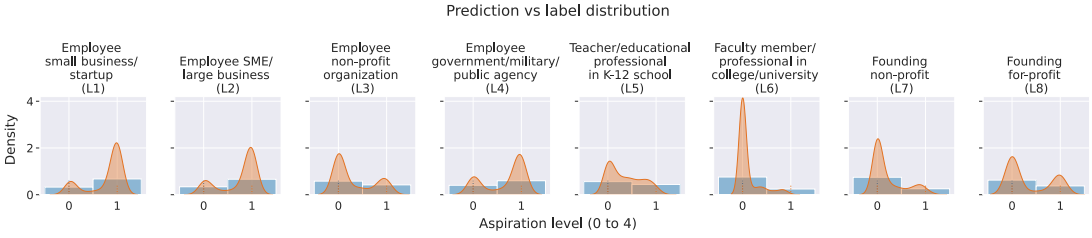
Table 5.3.: Macro f1 score in % for the different model combinations of text and numerical feature inputs with classification and regression head each. Results are presented for all of the 8 labels.



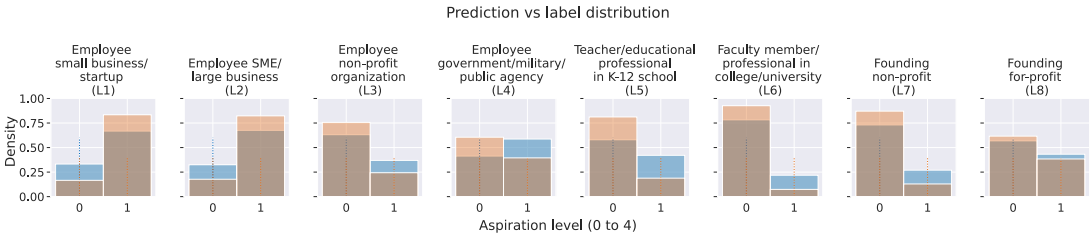
(a) Regression

(b) Classification

Figure 5.1.: Confusion matrices for regression and classification for the best model



(a) Regression



(b) Classification

Figure 5.2.: Distribution prediction and true labels

for the first set of feature attribution results. It gives the best performance and is the one we are most interested in to identify what sparks entrepreneurial aspirations for students. For the concept-based explanations, we will examine all of the heads since the concepts we extract present the information captured by the model in general.

5.2.1. Low-level Feature and Neuron Explanations with SHAP

As described in section 4.2.1, we first measure the global importance of numerical features and embeddings for the model's prediction. Figure 5.3 shows the global SHAP values for the top 10 features for prediction head 8. For a detailed description of which questions these features encode, refer to the table A.1 in the appendix A.

On the x-axis, one can see the SHAP values, which characterize the influence on the model prediction. The higher in magnitude the value is, the more important a feature is for the model, while a positive value contributes to a prediction value of 1 and a negative value to a class value of 0. The color furthermore shows the value of the feature. It, therefore, visualizes when there is a correlation between the feature value and its influence on the model prediction. There are two main observations we can derive from figure 5.3a.

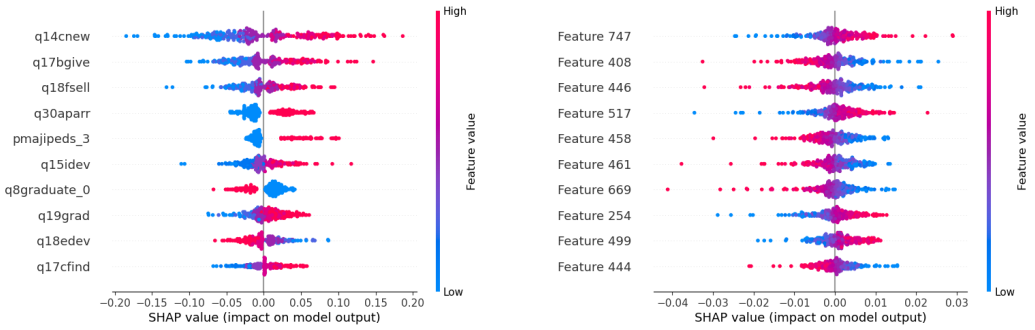
First, since all of the top 10 features are numerical features and no embedding neurons, we can clearly say that the model heavily relies on the information from the numerical answers instead of the information from the text to make predictions for the career goal. This also aligns with the observation we made for the results in section 5.1.3: Adding the text answers to the model input only led to a slight improvement in accuracy over the model just using the numerical features as input. Comparing the SHAP values for figure 5.3a with the overall top 10 features to figure with the top 10 embedding dimensions emphasizes this observation as the magnitude of the maximum SHAP values for the former is 0.2 versus 0.03 for the latter.

The second observation we can make from figure 5.3a is that the feature values highly correlate with the SHAP values, meaning the numerical features have a clear effect on the career goal. The three most important features are *q14cnew*, *q17bgive*, *q18fsell*, which are the variable names for the questions "In the past year, how often have you discussed *new design or business ideas* with other students?", "How much interest do you have in giving an *'elevator pitch' or presentation to a panel of judges about a new product or business*" and "How important is it to you to be involved in *selling a product or service in the marketplace* in the first five years after you graduate?" respectively, all having a Likert-scale answer scheme. Looking at these questions, it gets clear that a high feature value strongly correlates with entrepreneurial behavior, which is exactly what the SHAP values reflect. This means that the model takes this as an essential indicator for entrepreneurship. In general, most of the top 10 important features are related to

entrepreneurship with either the capability of building a product (*q15idev*) or bringing ideas to life (*q17cfind*, *q18edev*).

Also interesting is the 4th most important feature *q30aparr*, asking "Who in your life has had experience with starting their own business or organization?" and yes or no for "parents/guardians" as the answer to that question. The question, how much a person's background influences their interest in founding a startup has been heavily discussed in literature, with Lindquist, Sol, and Van Praag (2015) claiming that entrepreneurial parents increase their children's probability of becoming an entrepreneur by 60%. This also matches the claim that predetermined factors like social networks heavily influence students' entrepreneurial career aspirations (Eesley and Y. Wang 2017).

The top 10 essential features cover all of the topics except topic 3. This means that the model does not use the topic structure induced by the SCCT framework Lent, Brown, and Hackett (1994) by emphasizing one of the topics, but more selectively takes specific features form each topic to make its predictions.



(a) Embedding and numerical feature inputs

(b) Embedding input

Figure 5.3.: Global SHAP values describing the impact of the embedding layer and numerical feature inputs on the model's prediction

Beyond global explanations, we also want to get more detailed explanations locally on specific predictions as described in the methods section 4.2.1. For local explanations, we chose four samples. For these samples, we will not only look at the SHAP values for the numerical features and embedding level input (experiment 1 described in section 4.2.1), but also examine the text input's influence on the embedding layer and the prediction head (experiment 2 and 3 described in section 4.2.1). The local explanation results for experiment 1 are displayed for all of the four samples in figure 5.4 as a so-called force plot. They again show the SHAP values for the embeddings and the numerical features. The colors indicate whether the feature *pulls* the prediction in a positive (pink for class 1) or negative (blue for class 0) direction. The width of the

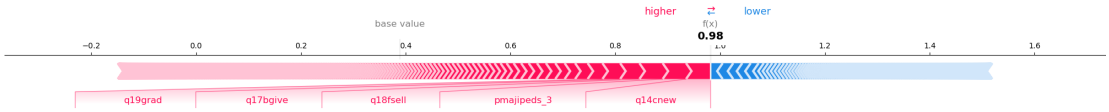
feature correlates with the magnitude or importance of this feature. The numerical features having the highest SHAP values for the local samples mostly overlap with the ones from the global explanation, which signals the explanation being coherent. Again the numerical features have higher SHAP values than the embeddings, being, therefore, more critical for the prediction.

Extending the explanation from the embedding layer to the text input layer with the second and third experiment described in section 4.2.1 gives us further insights on which parts of the language the model focuses on in order to make predictions for entrepreneurial aspirations. Figure 5.5 shows us the SHAP values for all the text input up to the same neuron of the embedding layer, which had the strongest influence on the prediction in experiment 1. We, therefore, see a general tendency of all the text input having high positive SHAP values for their contribution of the specific neurons in the embedding layer as they kind of maximize their activation.

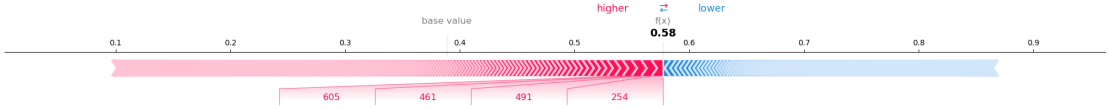
For understanding how the model extracts the idea of entrepreneurial aspirations from text, we must look at the SHAP values that show the influence of text input on the actual prediction. Figure 5.6 shows that for sample 1 belonging to class 1, "create something" and "start a business" gets most emphasized by the model, which can be fairly connected to entrepreneurship. For the second sample of class 1, the model focuses on "towards more sustainable solutions to our daily life", which means the model not only connects the process of creating solutions but also sustainability to entrepreneurship. Sample 3 belonging to class 0 has many concepts related to safety that the model interprets as not being related to entrepreneurship, as all of them push the model to a class prediction of 0. For sample 4 (class 0) the model again primarily focuses on the part of the sentences being connected to a steady life, which is "i'd love to be a desk-person [...] bury my face in paper work and be another employee [...] not looking to change the world [...] realism is best". From these examples, we get a clear understanding that the model uses parts in language speaking for or against entrepreneurial aspirations in a for humans understandable sense, while not just emphasizing abstract language syntax. However, it also does not reveal any hidden concept not detectable by humans. The parts emphasized by the model could have also been coded like this by a human. However, this means that the model's quality capturing the information provided in text reaches a human level but much more efficiently.

5.2.2. Higher-level Concept Explanations with ConceptSHAP

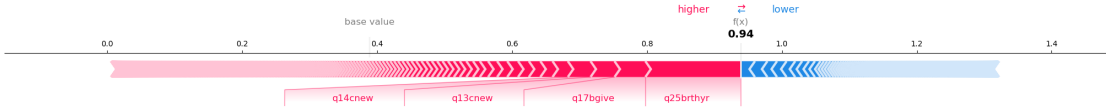
Having gained local insights on how the model processes text on a low-level input level, we now shed light on how it captures this input information globally in higher-level concepts and how it uses these concepts to make its predictions globally. We applied



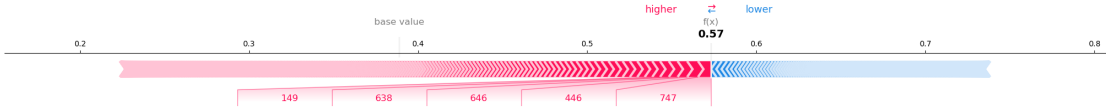
(a) Sample 1 belonging to class 1: Embedding and numerical feature inputs



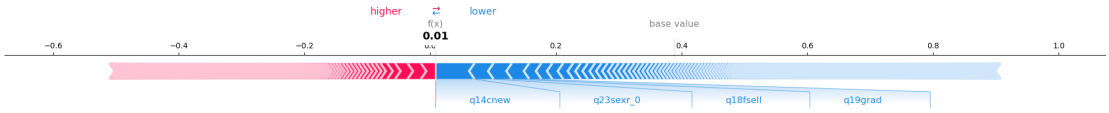
(b) Sample 1 belonging to class 1: Embedding input



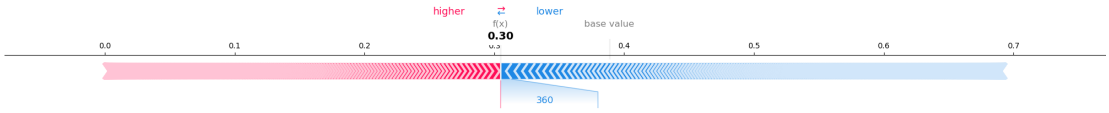
(c) Sample 2 belonging to class 1: Embedding and numerical feature inputs



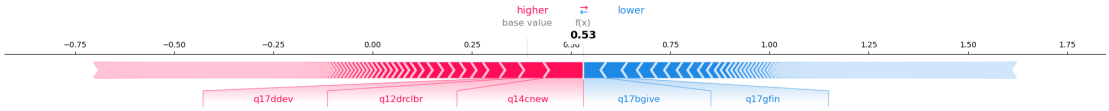
(d) Sample 2 belonging to class 1: Embedding input



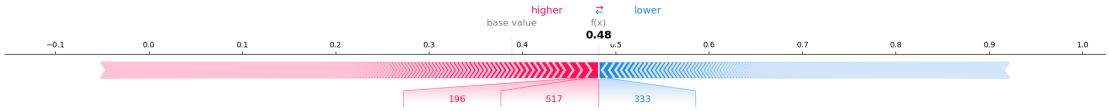
(e) Sample 3 belonging to class 0: Embedding and numerical feature inputs



(f) Sample 3 belonging to class 0: Embedding input



(g) Sample 4 belonging to class 0: Embedding and numerical feature inputs



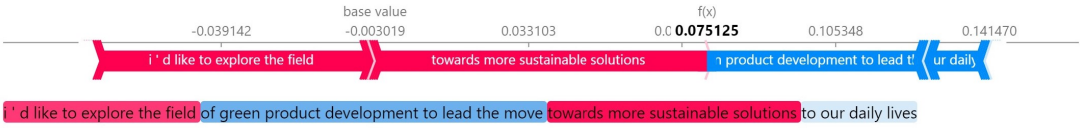
(h) Sample 4 belonging to class 0: Embedding input

Figure 5.4.: Local SHAP values describing the impact of the embedding layer and numerical feature inputs on the model's prediction for 4 different samples, 2 belonging to class 0 (not wanting to start a for-profit company) and 2 belonging to class 1 (wanting to start a for-profit company)

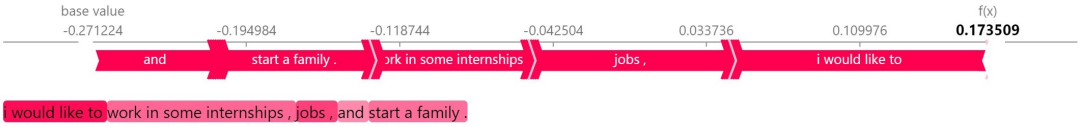
ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020) as explained in section 4.2.2 and set the number of concepts manually to four. We set the number quite low since we wanted to get broader, general concepts. Following (Yeh, B. Kim, Arik, et al. 2020), the concepts are presented twofold: First we look at the top 100 nearest neighbour word embeddings $\phi(x_t)$ having the highest concept score $v_c(x_t)$ belonging to a concept c_j. We then map the word token embeddings to the word tokens they belong to and take a broader piece (4 neighbor tokens) of the sentence instance they belong to. This gives us the topics presented in table 5.4 (only including the top 10 nearest neighbors).

We further count the word tokens appearing in the top 100 nearest neighbor sentence pieces and present the ones occurring more than five times, forming a word cloud as presented as well in table 5.4. The concepts being automatically extracted by the surrogate model can now be interpreted by humans finding a common theme in each of them. The first concept mainly contains nearest neighbors describing a lack of orientation and concrete career plans. This is also reflected by the word cloud being dominated by "no". The second concept, in contrast, captures a strong sense of future orientation and clear plans. Almost all the nearest neighbor samples start with "I", indicating strong self-centeredness being captured along the future orientation. This is also reflected by the word cloud, including "I" 63 out of 100 times. Also, other words like "plan" or "will" reflect the general future-oriented planning theme of this concept. Both concepts describe a certain "clarity of plans". This matches exactly one of the concepts extracted by Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken (2016, p. 8) through open-coding schemes.

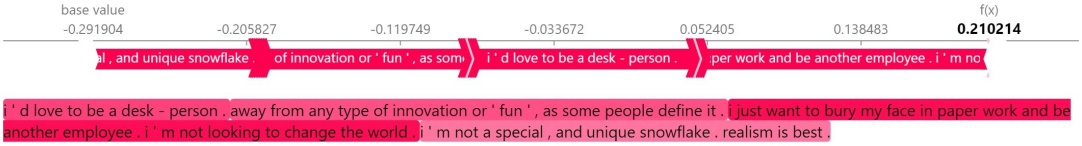
(a) Sample 1 belonging to class 1: Text input's influence on neuron 254 from the embedding layer



(b) Sample 2 belonging to class 1: Text input's influence on neuron 747 from the embedding layer



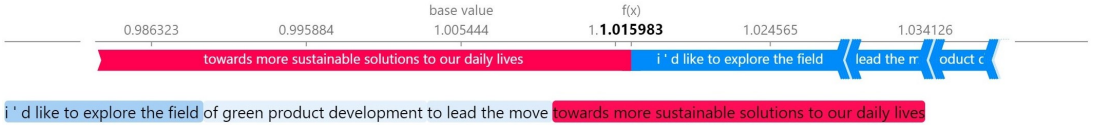
(c) Sample 3 belonging to class 0: Text input's influence on neuron 360 from the embedding layer



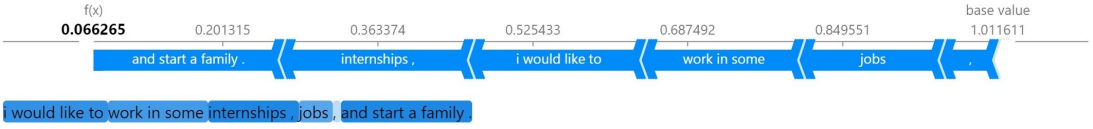
(d) Sample 4 belonging to class 0: Text input's influence on neuron 517 from the embedding layer

Figure 5.5.: Local SHAP values describing the impact of the text inputs on the neurons with the highest activation from the embedding layer explained in 5.4 for 4 different samples, 2 belonging to class 0 (not wanting to start a for-profit company) and 2 belonging to class 1 (wanting to start a for-profit company)

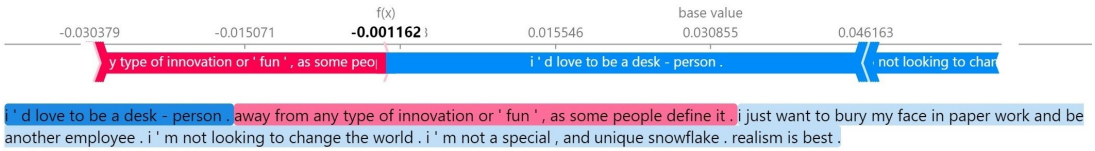
(a) Sample 1 belonging to class 1: Text input's influence on the prediction



(b) Sample 2 belonging to class 1: Text input's influence on the prediction



(c) Sample 3 belonging to class 0: Text input's influence on the prediction



(d) Sample 4 belonging to class 0: Text input's influence on the prediction

Figure 5.6.: Local SHAP values describing the impact of the text inputs on the prediction for 4 different samples, 2 belonging to class 0 (not wanting to start a for-profit company) and 2 belonging to class 1 (wanting to start a for-profit company)

While the first two concepts capture more the general idea of having a plan, the third concept reflects the different types of plans and career goals, containing nearest neighbors being related to the goal of founding a company, joining a startup, working in the industry, or pursuing a research career. The word cloud, therefore, mainly contains general words like "company", "work", or "engineering". Comparing this concept again to the concepts extracted through manual open-coding schemes shows again parallels to the code of "career characteristics" (Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken 2016, p. 8). The fourth concept is the most distinctive one as it does not capture ideas directly related to career goals. Being the dimension of time reflected by the nearest neighbors and the word cloud, this concept indirectly connects to career goals as being something that often includes some time-dimension that comes with planning and some timeline people follow in their career. This concept cannot be found among the codes from Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken (2016).

As mentioned in section 4.2.2, we are not only interested in the concepts qualitatively but also want to measure quantitatively whether they represent the information being captured by the model holistically. The completeness score measuring how well the surrogate model can recover each of the eight predictions from the original model according to equation 4.3 is presented in table 5.5. We see that for some heads, the completeness scores are negative, which means that the accuracy of the surrogate model is worse than the random accuracy. However, heads 6, 7, and 8 show high completeness scores, which indicates that the surrogate model can recover some predictions better than others. Since head eight has given the best results so far, receiving a sufficiently good completeness score of 0.73 for this head gives a coherent result. Both the original and the surrogate model can probably best extract some correlation in the data for this head.

The analysis so far was purely concept based analyzing how the model captures information in general, as described in section 2.4.3. As ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020) combines the concept-based approaches with causation analysis methods, we additionally want to present how the model uses these concepts to make predictions. Table 5.6 shows the ConceptSHAP values for each of the 8 prediction heads. The values in table 5.6 do not show any significant deviation between the concepts, which means that they all contribute about the same to the model's overall prediction.

To examine whether the concepts are just equally important for the model or also show no difference in how they spark the inner workings of the model, we ran one additional experiment for embedding activation patterns as described in 4.2.2. We thereby wanted to see whether the top 100 nearest neighbors within one topic activate the embedding layer in the same way but show different activation patterns between the different concepts. This follows the assumption, which Yeh, B. Kim, Arik, et al. (2020)

Concept	Nearest neighbours	Word cloud
1	want to be successful. find a job my own business no thanks work hard ill do whatever. no concrete plans yet run my own business. no comments no idea	software (5), my (6), no (17), thanks (6), idea (5), company (5), have (6), work (7)
2	i want to attend medical school i plan to find a mechanical i am planning to be a product i plan on working as a i would like to go into manufacturing and continue education with the eventual goal of i would first like to pursue doctoral degree having my own company i will be starting a career as an seeking law degree, to move into becoming	I (63), my (13), work (10), plan (24), find (5), graduate (8), will (17), be (17), go (7), am (5), career (6), get (6), job (7), would (13), like (14), engineering (7), working (13)
3	business learn skills, turn hobbies into i hope to run my own business start a company overseas earn experience in a small / .. either go into industry or go gain experience in the industry. would like to get into management own company when i have the expertise my feet in a start up company early a good paying job at a company that	company (19), my (13), industry (14), work (22), engineering (18), start (12), I (21), business (6), go (12), own (6), job (9), pursue (5), will (8), plan (6), engineer (5), get (7), degree (6), masters (5), working (13), be (5)
4	school within the next two years. work there for 3 years in the next five years i hope work abroad at some point. 5 to 6 years. at least the next two years, i there for at least three years. tentative at that point in time i want in the next five years i field at least once.	at (19), my (13), go (12), industry (14), work (22), engineering (18), start (12), I (21), business (6), engineer (5), be (5), own (6), job (9), pursue (5), will (8), plan (6), get (7), degree (6), masters (5), working (13)

Table 5.4.: The four concepts with 10 examples from the top 100 nearest neighbours and the word clouds containing the most frequent words from the nearest neighbours

L1	L2	L3	L4	L5	L6	L7	L8
-0.66	-0.79	0.17	-0.59	0.18	0.93	0.89	0.73

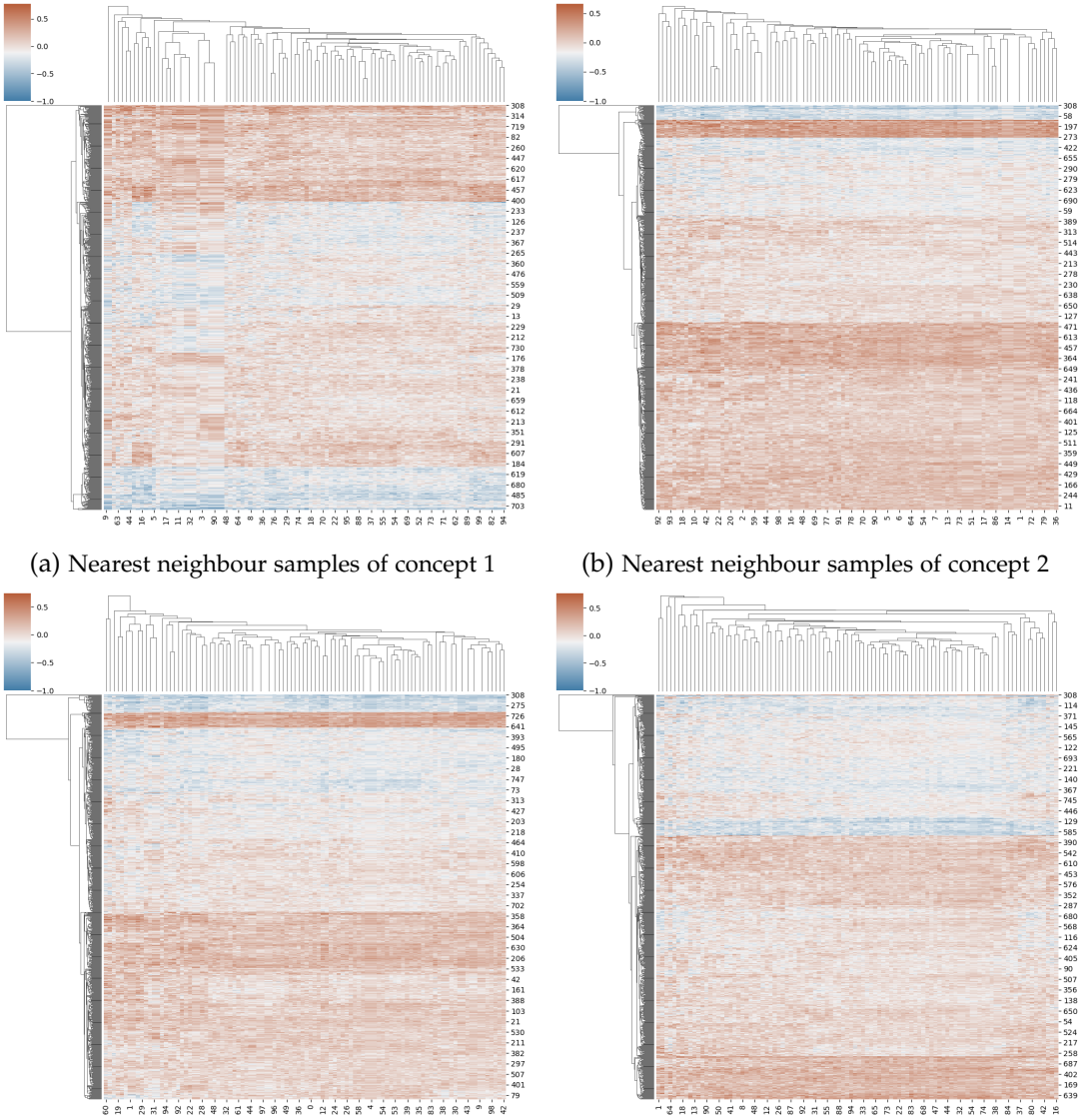
Table 5.5.: The completeness scores for each of the 8 prediction heads measuring how well the surrogate model can recover the prediction from the original model according to equation 4.3

Class	Concept	L1	L2	L3	L4	L5	L6	L7	L8
Overall	1	-0.16	-0.33	0.19	-0.15	0.04	0.23	0.22	0.18
	2	-0.16	-0.16	0.02	-0.14	0.04	0.22	0.22	0.18
	3	-0.16	-0.17	-0.08	-0.15	0.04	0.25	0.22	0.18
	4	-0.16	-0.13	0.03	-0.15	0.04	0.22	0.22	0.18

Table 5.6.: The ConceptSHAP values measuring the contribution of the 4 concepts to each of the 8 prediction heads/labels; showing the contribution to the model's overall prediction

used for the regularization term of the objective function for the surrogate model (see equation 4.5), stating that samples within the concept should be as similar as possible. However, different concepts should be as distinctive or "far" as possible. The difference, however, is that the regularization term focuses on the single word token embeddings while we study the mean embedding vectors for one whole sample sentence for this experiment.

The activation patterns are presented for each of the four different concepts in figure 5.7. We see a very homogeneous activation pattern for nearest neighbor samples within each of the concepts but different patterns from topic to topic, which matches our assumption that the concepts capture how the model processes different types of information.



(a) Nearest neighbour samples of concept 1

(b) Nearest neighbour samples of concept 2

(c) Nearest neighbour samples of concept 3

(d) Nearest neighbour samples of concept 4

Figure 5.7.: Activations of mean embedding vectors for samples belonging to each of the four concepts. Concept 0 to 4 from left to right. The y-axis belongs to the 768 embedding neurons, while the x-axis shows the 100 samples. The four concepts show a homogeneous activation pattern for samples within the concept but differ a lot for samples of different concepts.

6. Discussion and Conclusion

In this thesis, we introduced an approach to analyzing survey data from the Engineering Major Survey (EMS 1.0) using neural NLP and XAI methods.

Thereby, we answered two research questions, as introduced in chapter 1. By having a methodological as well as an application related focus, we split the overall problem statement of survey analysis into two parts in order to answer these two research questions.

For the first research question (RQ1) we built a deep neural model, processing both open- and closed-ended answers from the survey to predict the students' career goals. Taking both types of answers as raw inputs at the same time –- eliminating the need to pre-process the open-ended answers with coding schemes — already provides an improvement compared to traditional closed-vocabulary methods using dictionaries (Stone, Bales, Namenwirth, and Ogilvie 1962; Hart 1984; Pennebaker, Francis, and Booth 1999; Pennebaker, Booth, Boyd, and Francis 2015). Additionally, this approach offers two major benefits. Firstly, our method is free from human bias by processing the data automatically. Secondly, the model can learn to connect information between the open- and closed-ended answers to base its prediction on both.

We extensively tested different model architectures with different input variables in order to detect the most accurate prediction performance by using the macro f1 score (see section 5.1). While the results for most of the eight prediction heads related to different career perspectives show limited success, we were able to get reasonable f1 score results for the career goal of the eighth prediction head – representing entrepreneurial aspirations of "founding a for-profit company". We assume the small sample size and the low correlation of the dataset are the reasons for the model's limited predictive power. Application-wise we are most interested in entrepreneurship. Therefore, focusing our further methodology on the eighth prediction head still allowed us to get insights into which factors influence students' entrepreneurial aspirations.

In order to answer the second research question (RQ2), we applied two explainability methods SHAP (Lundberg and S.-I. Lee 2017) and ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020). With SHAP we are able to understand how the numerical features in comparison to the text embeddings contribute globally and locally to the model's predictions. This analysis revealed that numerical features have a much higher impact on the model than text embeddings. The major advantage of this method over

traditional quantitative analyses (e.g., Atwood, S. Gilmartin, Harris, and Sheppard (2021) and M. Schar, S. Gilmartin, Rieken, et al. (2017)) is the fact that we could analyze the influence of both open- and closed-ended answers at the same time. Instead of examining linear relationships with correlation analysis between selected variables (e.g., Atwood, S. Gilmartin, Harris, and Sheppard (2021)), we are able to explain highly nonlinear influences of all variables in the dataset on the prediction of the career variable at the same time – extracting the most important ones automatically. The most critical numerical variables extracted by this model are highly coherent with the human understanding of entrepreneurship (see section 5.2.1).

Regarding the explanation of how the open-ended questions correlate with the career goal variable we are able to understand how the model processes language locally on a direct input level with SHAP (Lundberg and S.-I. Lee 2017) and globally on a more abstract level with ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020).

Both the local text explanations and the concepts delivered coherent results for features and concepts that strongly reflect our human understanding of entrepreneurship and career goals. Another finding suggests strong parallels between the automatically concepts extracted with ConceptSHAP and the manually assigned codes by Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken (2016). While three of four concepts exactly match the manual codes, the fourth concept extracts an additional latent factor to the ones proposed with manual methods.

However, ConceptSHAP did not deliver a sufficient explanation for the concepts' contribution to the model's overall prediction. Hence, we further examined whether the model processes samples belonging to the concepts differently and found that samples within their concepts show similar embedding activation patterns while being distinctive from other concepts. This implies that the concepts give a good representation of how the model processes different types of information from text.

The combined analysis of SHAP and ConceptSHAP delivers a thorough understanding of how the model extracts patterns in language to predict a career goal. This distinguishes our approach from existing methods like topic models from open-vocabulary methods which deliver less expressive concept explanations on a word cloud level based on a bag-of-word models (Eichstaedt, Kern, Yaden, et al. 2020; Deerwester, Dumais, Furnas, et al. 1990; Blei, Ng, and Jordan 2003). In contrast, our explanations reveal highly nonlinear relations extracted by a complex model – capturing spatial semantics of text. In particular, the method of ConceptSHAP (Yeh, B. Kim, Arik, et al. 2020) entails the benefit of extracting the concepts in an unsupervised fashion, as opposed to other concept methods (B. Kim, Wattenberg, Gilmer, et al. 2018; Ghorbani, Wexler, J. Y. Zou, and B. Kim 2019). Given the strong results of concept identification similar to manual coding, unsupervised concept extraction promises a huge potential for its application in survey analysis. The automatically extracted concepts could be

either used as a replacement for manual coding or as an initial coding scheme, which humans can further refine.

Application-wise, our findings for the numerical features show that interest and experience in business- product-centered activities have the strongest influence on the entrepreneurial aspirations. Beyond that, students' relation to entrepreneurship is strongly influenced by their parents as entrepreneurial role models. Whereas the former extracted features related to entrepreneurial activities guide entrepreneurship education design, the latter background feature reveals the importance of a student's access to entrepreneurial role models. Eesley and Y. Wang (2017) already investigates mentor-ship as alternative access for students without touch-points to entrepreneurship. Moreover, it is interesting that our model captures how students integrate the dimension of time in their career goals in one of the extracted concepts. Park and Jung (2015) also studied the time dimension in career planning in terms of the *Future Time Perspective* (FTP). This is a measure for someone thinking of their future as being time-limited. They found out that FTP directly correlates with career commitment.

In conclusion, this thesis proves that methods from NLP and XAI are suitable tools to automatically analyze survey data of open- and closed-ended questions. Beyond saving researchers' time and resources, this methodology is also able to provide more profound insights into the data than traditionally used closed-vocabulary or quantitative methods from social sciences and open-vocabulary methods from computer science.

7. Future Work

We proposed an approach for profoundly and innovatively analyzing surveys with new XAI and NLP methods. We showed that these methods find suitable application in survey analysis. However, there are still some shortcomings and open questions that can be addressed by future researchers.

The most significant limitation of our model is the low performance of the applied prediction model, being traced back to the small size and limited expressiveness of the dataset. It is worth exploring models to overcome this disadvantage since a small data sample size is quite common for surveys in social sciences.

One further research direction would be a more extensive exploration of ConceptSHAP - being a promising research direction in the field of XAI. One approach could be to study concept explanations on a local, and not only on a global level. A first analysis for this local approach could connect a sample with the extracted concepts by assigning each concept a score or probability of belonging to the overall sample instead of linking it to a single word token. This would reduce the dimension of the sample from the embedding vector size ($\in \mathbb{R}^{768}$) to the concept vector size ($\in \mathbb{R}^{5}$) and show how prevalent the extracted concepts are within one sample. This is similar to open- and closed-vocabulary methods described in section 2.2. One could, for example, compute this score or probability similar to the frequency score in section 2.2.1 as the highest concept score for each word token, and then calculate the share of frequency it scores highest. Another variant of assigning concepts to the overall sample rather than only to word tokens would be to compute the mean concept score for each concept across all word tokens, similar to how the sample embedding vector is computed from the word token embedding vectors. In addition to this text-based perspective of assigning concepts to samples, researchers could also compute ConceptSHAP values for a single prediction to extract the most influential concepts for a sample's prediction. Comparing both would give us an idea of whether the model uses a concept only because it is the most predominant concept in the text (if the results were similar) or whether it also focuses on concepts not being strongly presented in the sample.

One further research direction would be to explore the idea of ConceptSHAP for features, trying to extract concepts consisting of certain features instead of words. This could further be combined with text concepts. Since the number of concepts must be set manually in our experiments, it could also be worth exploring an automated way

of setting the number of concepts as a hyperparameter. This can then be tuned with automated methods using the completeness score as a performance measurement.

We introduce ConceptSHAP as an alternative to traditional, closed-vocabulary methods like manual coding or open-vocabulary methods. Therefore, performing a comparison analysis with different applications would reveal further insights into the similarities and differences and provide some guidance on suitable applications to use each of them.

On the application side, we found the consideration of time as a latent concept captured by the model. This suggests to include this variable as a factor for career goals, in further research. In addition, other concepts capturing different forms of clarity in career plans give new research direction of whether a strong tendency towards planning is a personality trait that influences career decisions.

A. Appendix

This appendix shows a table of the dataset we use, including its labels, the text feature variables and the numerical feature variables.

Topic	Variable	Description
	prediction label	
job targets	q20: How likely is it that you will do each of the following in the first five years after you graduate?	
	q20asbus	Work as an employee for a small business or start-up company
	q20blbus	Work as an employee for a medium- or large-size business
	q20cnon	Work as an employee for a non-profit organization (excluding a school or college/university)
	q20dgov	Work as an employee for the government, military, or public agency (excluding a school or college/university)
	q20etch	Work as a teacher or educational professional in a K-12 school
	q20fcoll	Work as a faculty member or educational professional in a college or university
	q20hsnon	Found or start your own for-profit organization
	q20gsfor	Found or start your own non-profit organization
	text input variable	
text topic	q22career	We have asked a number of questions about your future plans. If you would like to elaborate on what you are planning to do, in the next five years or beyond, please do so here.
	inspire	To what extent did this survey inspire you to think about your education in new or different ways? Please describe.

Topic	Variable	Description
numerical feature input variable		
topic 1	q9: During high school, did you:	
	q9aart	Take an art, dance, music, theater, or creative writing class
	q9bcomp	Learn computer programming
	q9cshop	Take a shop class (e.g., a woodworking, automotive, or maker class) or engineering class
	q9drobo	Participate in a robotics competition, such as a FIRST Robotics Competition
	q9ecamp	Attend a science, math, technology, or engineering related summercamp
	q9fres	Have a research position or internship at a science, math, technology, or engineering related company or organization
	q9geship	Learn about entrepreneurship
	q9hstart	Start or co-found your own club, organization, or company
	q10: While an undergraduate, have you done (or are you currently doing) each of the following for at least one full academic or summer term?	
	q10ares	Conduct research with a faculty member
	q10bint	Work in a professional engineering environment as an intern/co-op
	q10cpay	Have a work-study or other type of job to help pay for your college education
	q10dabrd	Participate in study abroad
	q11: As part of your undergraduate coursework so far (including courses you are currently taking), have you taken courses that include any of the following topics or components?	
	q11aart	Art, dance, music, theater, or creative writing
	q11bcomp	Computer science
	q11ctheo	Theory of design
	q11ddes	Designing and/or prototyping things or ideas
	q11ebus	Business or enterprise topics (including entrepreneurship or venture creation)
	q11flead	Leadership topics
	q11ginter	Interaction with students from non-engineering majors

Topic	Variable	Description
numerical feature input variable		
topic 1	q12: Below are various extra- and co-curricular activities you may have been involved in while an undergraduate. Many of these have to do with innovation and/or entrepreneurship; others are more general to the college experience. Please mark which of the following you have done during your undergraduate years so far (including activities you are currently doing).	
	q12abclbr	A business or entrepreneurship club
	q12bcclbr	A community service-based club
	q12cdclbr	A design club
	q12drclbr	A robotics club
	q12eeclbr	Other student clubs or groups in engineering
	q12fnclbr	Other student clubs or groups outside of engineering
	q12gbcomr	A business plan, business model, or elevator pitch competition
	q12hdcomr	A design or invention competition
	q12iscomr	A social entrepreneurship/social innovation competition
	q12jspcer	A maker space/design or inventors studio/prototyping lab
	q12kcarr	A career related event or meeting (e.g., a college career fair, a one-on-one meeting with a career counselor)
	q12lspkr	A speaker series or related presentations about entrepreneurship and/or innovation
	q12mbootr	A start-up bootcamp (e.g., Start-up Weekend, 3-Day Startup)
	q12npresr	A presentation on a new engineering technology, process, or design (outside of class)
	q12oedrmr	A residential or dorm-based engineering program/engineering living-learning community
	q12pidrmr	A residential or dorm-based entrepreneurship or innovation program/entrepreneurship or innovation living learning community
	q12qfundr	Funding from a program to finance new ideas
	q12rleadr	A student organization
	q12sstsr	A student club or other student group on campus
	q12tstor	Your own for-profit or non-profit organization
	q12unoner	None of the above
	q12vnoanr	I prefer not to answer

Topic	Variable	Description
numerical feature input variable		
topic 1	q13: In the past year, how often have you discussed each of the following with faculty members at your institution?	
	q13acrs	Course topics and assignments (not during class or section time)
	q13bopt	Your professional options with an engineering degree
	q13cnew	New design or business ideas
	q14: In the past year, how often have you discussed each of the following with other students?	
	q14acrs	Course topics and assignments (not during class or section time)
	q14bopt	Your professional options with an engineering degree
	q14cnew	New design or business ideas
topic 2	q15: How confident are you in your ability to do each of the following at this time?	
	q15aask	Ask a lot of questions
	q15bgen	Generate new ideas by observing the world
	q15cexp	Experiment as a way to understand how things work
	q15dact	Actively search for new ideas through experimenting
	q15ebld	Build a large network of contacts with whom you can interact to get ideas for new products or services
	q15fcon	Connect concepts and ideas that appear, at first glance, to be unconnected
	q15gdes	Design a new product or project to meet specified requirements
	q15hcdt	Conduct experiments, build prototypes, or construct mathematical models to develop or evaluate a design
	q15idev	Develop and integrate component sub-systems to build a complete system or product
	q15jana	Analyze the operation or functional performance of a complete system
	q15ktrb	Troubleshoot a failure of a technical component or system
	q15llead	Lead a team of people
	q15mcomm	Communicate your ideas effectively to people in different positions or fields
	q15ntake	Take the steps needed to place a financial value on a new business venture

Topic	Variable	Description
numerical feature input variable		
topic 3	q16: Imagine the work you will be doing in the first year after you graduate. Estimate what will happen if you "ask a lot of questions" in this work.	
	q16astar	I will be seen as a "star" in this work.
	q16bresp	will earn the respect of my colleagues.
	q16churt	I will hurt my chances for moving ahead.
	q16dtrbl	I will be seen as a troublemaker.
topic 4	q23sexr	What is your sex?
	q24: What is your racial or ethnic identification?	
	q24aaminr	American Indian or Alaska Native
	q24basamr	Asian or Asian American
	q24cafamr	Black or African American
	q24dlatr	Hispanic or Latino/a
	q24enhpir	Native Hawaiian or Pacific Islander
	q24fwhtr	White
	q24other	Other (please specify):
	URM	Underrepresented Racial/Ethnic Minority
	q25brthyr	In which year were you born?
	q25brthyr96m	flag for students born in 1996 or earlier
	q26cit	Are you: A United States Citizen/ A Permanent Resident of the United States/ Other
	q27born	Were you born in the U.S.?
	q28immi	Did one or more of your parents/guardians immigrate to the U.S.?
	q29fam	When you were growing up, would you describe your family as: Low income - High income
	q30: Who in your life has had experience with starting their own business or organization?	
	q30aparr	Parents/guardians
	q30bsibr	Siblings
	q30crelr	Other relatives
	q30dfrndr	Other friends or contacts
	q30enoner	None of the above (no one)
	q32par1ed	How much education did your "Parent 1" complete? Did not finish high school - Completed a Doctoral or Professional degree (JD, MD, PhD, etc.)
	q34par2ed	How much education did your "Parent 1" complete? Did not finish high school - Completed a Doctoral or Professional degree (JD, MD, PhD, etc.)

Topic	Variable	Description
numerical feature input variable		
topic 5	q17: How much interest do you have in:	
	q17aexp	Experimenting in order to find new ideas
	q17bgive	Giving an "elevator pitch" or presentation to a panel of judges about a new product or business idea
	q17cfind	Finding resources to bring new ideas to life
	q17ddev	Developing plans and schedules to implement new ideas
	q17ecdt	Conducting basic research on phenomena in order to create new knowledge
	q17fsoc	Working on products, projects, or services that address societal challenges
	q17gfin	Working on products, projects, or services that have significant financial potential
topic 6	q18: How important is it to you to be involved in the following job or work activities in the first five years after you graduate?	
	q18asrch	Searching out new technologies, processes, techniques, and/or product ideas
	q18bgen	Generating creative ideas
	q18cprom	Promoting and championing ideas to others
	q18dinve	Investigating and securing resources needed to implement new ideas
	q18edev	Developing adequate plans and schedules for the implementation of new ideas
	q18fsell	Selling a product or service in the marketplace
topic 7	q19atyp1r	How likely is it that you will enter graduate school in the first five years after you graduate?
	q19atyp1r	Degree 1: Type
	q19afld1r	In which engineering field?
	q21: How likely is it that your work will involve engineering (e.g., engineering practice, research, management, or sales) in...	
	q21a1y	The first year after you graduate
	q21b5y	Five years after you graduate
	q21c10y	Ten years after you graduate

Topic	Variable	Description
numerical feature input variable		
topic 8	q2class	What is your current academic standing?
	q3status	Are you enrolled primarily as a full-time/part-time student
	q4transfer	Did you transfer to your current institution from another college/university?
	q5aptype	In this major, are you pursuing a BS/BA
	q5bpconr	Are you pursuing a concentration or topical track within this major?
	q5cpcomp	Do you intend to complete this major?
	q6second	Are you pursuing a second (double) major?
	q7: In addition to your major(s), are you pursuing a minor or certificate for academic credit?	
	q7anor	no
	q7bminr	Yes, a minor
	q7ccertr	Yes, a certificate
	q8graduate	During which year do you anticipate graduating from your current institution with your bachelor's degree(s)?
	q35gpa	What is your overall college grade point average?

Table A.1.: The dataset with its label variable, the 2 text input variables and the 119 numerical feature variables.

List of Figures

List of Tables

Bibliography

[1] G. Alain and Y. Bengio. *Understanding intermediate layers using linear classifier probes*. 2016.

[2] J. Alammar. *Interfaces for Explaining Transformer Language Models*. 2020.

[3] J. Alammar. *The Illustrated Transformer*. Dec. 2018.

[4] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. "Gradient-Based Attribution Methods." In: *Explainable AI*. 2019.

[5] L. Arras, G. Montavon, K.-R. Müller, and W. Samek. "Explaining Recurrent Neural Network Predictions in Sentiment Analysis." In: Sept. 2017. DOI: 10.18653/v1/W17-5221.

[6] L. Arras, A. Osman, K.-R. Müller, and W. Samek. *Evaluating Recurrent Neural Network Explanations*. 2019.

[7] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, R. Chatila, and F. Herrera. *Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI*. 2019.

[8] P. Atanasova, J. G. Simonsen, C. Lioma, and I. Augenstein. *A Diagnostic Study of Explainability Techniques for Text Classification*. 2020.

[9] S. Atwood, S. Gilmartin, A. Harris, and S. Sheppard. "Defining First-generation and Low-income Students in Engineering: An Exploration." In: *2020 ASEE Virtual Annual Conference Content Access Proceedings*. ASEE Conferences, 2021. DOI: 10.18260/1-2--34373.

[10] M. Aubakirova and M. Bansal. *Interpreting Neural Networks to Improve Politeness Comprehension*. 2016.

[11] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation." In: *PloS one* 10.7 (2015), e0130140. DOI: 10.1371/journal.pone.0130140.

[12] D. A. Bau*, Y. Belinkov*, H. Sajjad, F. Dalvi, N. Durrani, and J. Glass. "Identifying and Controlling Important Neurons in Neural Machine Translation." In: *International Conference on Learning Representations (ICLR)*. New Orleans, US, May 2019.

[13] S. Bhatia. "Associative judgment and vector space semantics." In: *Psychological review* 124.1 (2017), pp. 1–20. DOI: 10.1037/rev0000047.

[14] D. M. Blei, A. Y. Ng, and M. I. Jordan. "Latent Dirichlet Allocation." In: *J. Mach. Learn. Res.* 3.null (Mar. 2003), pp. 993–1022. ISSN: 1532-4435.

[15] H. Brandstätter. "Personality aspects of entrepreneurship: A look at five meta-analyses." In: *Personality and Individual Differences* 51.3 (2011), pp. 222–230. ISSN: 01918869. DOI: 10.1016/j.paid.2010.07.007.

[16] A. Bryant and K. Charmaz. *The Sage handbook of grounded theory*. Sage, 2007.

[17] F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, A. Bau, and J. Glass. *What Is One Grain of Sand in the Desert? Analyzing Individual Neurons in Deep NLP Models*. 2018.

[18] F. Dalvi, H. Sajjad, N. Durrani, and Y. Belinkov. "Analyzing Redundancy in Pretrained Transformer Models." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 4908–4926. DOI: 10.18653/v1/2020.emnlp-main.398.

[19] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen. "A Survey of the State of Explainable AI for Natural Language Processing." In: *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing* ().

[20] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. "Indexing by latent semantic analysis." In: *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE* 41.6 (1990), pp. 391–407.

[21] M. Denil, A. Demiraj, and N. de Freitas. *Extraction of Salient Sentences from Labelled Documents*. 2014.

[22] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 4171–4186. DOI: 10.18653/v1/N19-1423.

[23] J. Dyer, H. Gregersen, and C. M. Christensen. *Innovator's DNA, Updated, with a New Preface: Mastering the Five Skills of Disruptive Innovators*. Harvard Business Press, 2019.

[24] C. Eesley and Y. Wang. "Social influence in career choice: Evidence from a randomized field experiment on entrepreneurial mentorship." In: *Research Policy* 46.3 (2017), pp. 636–650. ISSN: 0048-7333. DOI: https://doi.org/10.1016/j.respol.2017.01.010.

[25] j. C. Eichstaedt, M. L. Kern, D. B. Yaden, H. A. Schwartz, S. Giorgi, G. Park, C. Hagan, V. Tobolsky, L. K. Smith, A. Buffone, J. Iwry, M. Seligman, and L. H. Ungar. *Closed and Open Vocabulary Approaches to Text Analysis: A Review, Quantitative Comparison, and Recommendations*. 2020. DOI: 10.31234/osf.io/t52c6.

[26] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. *Visualizing Higher-Layer Features of a Deep Network*. Tech. rep. 1341. Also presented at the ICML 2009 Workshop on Learning Feature Hierarchies, Montréal, Canada. University of Montreal, June 2009.

[27] R. Fischer. "Standardization to Account for Cross-Cultural Response BiasA Classification of Score Adjustment Procedures and Review of Research in JCCP." In: *Journal of Cross-cultural Psychology - J CROSS-CULT PSYCHOL* 35 (May 2004), pp. 263–282. DOI: 10.1177/0022022104264122.

[28] D. Freedman, R. Pisani, and R. Purves. "Statistics (international student edition)." In: *Pisani, R. Purves, 4th edn. WW Norton & Company, New York* (2007).

[29] S. Freud. *Psychopathology of everyday life*. Vol. 24. Penguin Group, 1938.

[30] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim. "Towards Automatic Concept-based Explanations." In: *NeurIPS*. 2019.

[31] A. Ghorbani and J. Zou. *Neuron Shapley: Discovering the Responsible Neurons*. 2020.

[32] S. K. Gilmartin, H. L. Chen, M. F. Schar, Q. Jin, G. Toye, A. Harris, E. Cao, E. Costache, M. Reithmann, and S. D. Sheppard. "Designing a longitudinal study of engineering students' innovation and engineering interests and plans: The Engineering Majors Survey Project. EMS 1.0 and 2.0 Technical Report." In: *Stanford University Designing Education Lab, Stanford, CA, Technical Report* (2017).

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik. *Rich feature hierarchies for accurate object detection and semantic segmentation Supplementary material*. 2014.

[34] D. Gopinath, H. Converse, C. S. Păsăreanu, and A. Taly. "Property Inference for Deep Neural Networks." In: *Proceedings of the 34th IEEE/ACM International Conference on Automated Software Engineering*. ASE '19. San Diego, California: IEEE Press, 2019, pp. 797–809. ISBN: 9781728125084. DOI: 10.1109/ASE.2019.00079.

[35] A. Graves and J. Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." In: *Neural Networks* 18.5 (2005). IJCNN 2005, pp. 602–610. ISSN: 0893-6080. DOI: https://doi.org/10.1016/j.neunet.2005.06.042.

[36] E. A. Greenleaf. "Improving Rating Scale Measures by Detecting and Correcting Bias Components in Some Response Styles." In: *Journal of Marketing Research* 29.2 (1992), pp. 176–188. DOI: 10.1177/002224379202900203. eprint: https://doi.org/10.1177/002224379202900203.

[37] T. Guetterman, T. Chang, M. DeJonckheere, T. Basu, E. Scruggs, and V. Vydiswaran. "Augmenting Qualitative Text Analysis with Natural Language Processing: Methodological Study." In: *Journal of Medical Internet Research* 20 (June 2018), e231. DOI: 10.2196/jmir.9702.

[38] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, D. Pedreschi, and F. Giannotti. *A Survey Of Methods For Explaining Black Box Models*. 2018.

[39] R. P. Hart. "Systematic analysis of political discourse: The development of DICTION." In: *Political communication yearbook* 1 (1984), pp. 97–134.

[40] J. Hewitt and P. Liang. "Designing and Interpreting Probes with Control Tasks." In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2019, pp. 2733–2743. DOI: 10.18653/v1/D19-1275.

[41] J. Howard and S. Ruder. "Universal Language Model Fine-tuning for Text Classification." In: *ACL*. 2018.

[42] F. Jafariakinabad, S. Tarnpradab, and K. Hua. "Syntactic Recurrent Neural Network for Authorship Attribution." In: (Feb. 2019).

[43] M. Jayaratne and B. Jayatilleke. "Predicting Personality Using Answers to Open-Ended Interview Questions." In: *IEEE Access* 8 (2020), pp. 115345–115355. DOI: 10.1109/ACCESS.2020.3004002.

[44] K. S. Jones. "A statistical interpretation of term specificity and its application in retrieval." In: *Journal of Documentation* 28 (1972), pp. 11–21.

[45] Á. Kádár, G. Chrupała, and A. Alishahi. *Representation of linguistic form and function in recurrent neural networks*. 2016.

[46] R. Kanter. "Chapter 7. When a Thousand Flowers Bloom: Structural, Collective, and Social Conditions for Innovation in Organizations." In: *Research in Organizational Behavior* 10 (July 2013). DOI: 10.1016/B978-0-7506-9749-1.50010-7.

[47] A. Karpathy, J. Johnson, and F.-F. Li. "Visualizing and Understanding Recurrent Networks." In: *Cornell Univ. Lab.* (June 2015).

[48] A. Katz, M. Norris, A. M. Alsharif, M. D. Klopfer, D. B. Knight, and J. R. Grohs. "Using Natural Language Processing to Facilitate Student Feedback Analysis." In: *2021 ASEE Virtual Annual Conference Content Access*. 2021.

[49] B. Kim, M. Wattenberg, J. Gilmer, C. J. Cai, J. Wexler, F. Viégas, and R. Sayres. "Interpretability Beyond Feature Attribution: Quantitative Testing with Concept Activation Vectors (TCAV)." In: *ICML*. 2018.

[50] P. W. Koh and P. Liang. *Understanding Black-box Predictions via Influence Functions*.

[51] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. E. Barnes, and D. E. Brown. *Text Classification Algorithms: A Survey*. 2019.

[52] C. Kramsch and H. Widdowson. *Language and culture*. Oxford university press, 1998.

[53] Y. Lakretz, G. Kruszewski, T. Desbordes, D. Hupkes, S. Dehaene, and M. Baroni. "The emergence of number and syntax units in LSTM language models." In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, June 2019, pp. 11–20. DOI: 10.18653/v1/N19-1002.

[54] E. P. Lazear. "Balanced Skills and Entrepreneurship." In: *American Economic Review* 94.2 (May 2004), pp. 208–211. DOI: 10.1257/0002828041301425.

[55] Q. V. Le. "Building high-level features using large scale unsupervised learning." In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*. Piscataway, NJ: IEEE, 2013, pp. 8595–8598. ISBN: 978-1-4799-0356-6. DOI: 10.1109/ICASSP.2013.6639343.

[56] M. D. LeCompte. "Analyzing qualitative data." In: *Theory into practice* 39.3 (2000), pp. 146–154.

[57] W. Leeson, A. Resnick, D. Alexander, and J. Rovers. "Natural Language Processing (NLP) in Qualitative Public Health Research: A Proof of Concept Study." In: *International Journal of Qualitative Methods* 18 (2019), p. 1609406919887021. DOI: 10.1177/1609406919887021.

[58] R. W. Lent, S. D. Brown, and G. Hackett. "Toward a Unifying Social Cognitive Theory of Career and Academic Interest, Choice, and Performance." In: *Journal of Vocational Behavior* 45.1 (1994), pp. 79–122. ISSN: 00018791. DOI: 10.1006/jvbe.1994.1027.

[59] A. Levine, T. Björklund, S. Gilmartin, and S. Sheppard. "A Preliminary Exploration of the Role of Surveys In Student Reflection and Behavior." In: *2017 ASEE Annual Conference & Exposition Proceedings*. ASEE Conferences, 2017. DOI: 10.18260/1-2--27500.

[60] J. Li, X. Chen, E. Hovy, and D. Jurafsky. *Visualizing and Understanding Neural Models in NLP*. 2015.

[61] J. Li, W. Monroe, and D. Jurafsky. *Understanding Neural Networks through Representation Erasure*. 2016.

[62] M. J. Lindquist, J. Sol, and M. Van Praag. "Why do entrepreneurial parents have entrepreneurial children?" In: *Journal of Labor Economics* 33.2 (2015), pp. 269–296.

[63] 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*. 2019.

[64] Z. Liu, Y. Lin, and M. Sun. "Representation Learning and NLP." In: *Representation Learning for Natural Language Processing*. Singapore: Springer Singapore, 2020, pp. 1–11. ISBN: 978-981-15-5573-2. DOI: 10.1007/978-981-15-5573-2_1.

[65] S. Lundberg and S.-I. Lee. *A Unified Approach to Interpreting Model Predictions*. 2017.

[66] S. M. Mathews. "Explainable Artificial Intelligence Applications in NLP, Biomedical, and Malware Classification: A Literature Review." In: *Intelligent Computing*. Ed. by K. Arai, R. Bhatia, and S. Kapoor. Cham: Springer International Publishing, 2019, pp. 1269–1292. ISBN: 978-3-030-22868-2.

[67] M. Mehl, S. Gosling, and J. Pennebaker. "Personality in its natural habitat: Manifestations and implicit folk theories of personality in daily life." In: *Journal of personality and social psychology* 90 (June 2006), pp. 862–77. DOI: 10.1037/0022-3514.90.5.862.

[68] R. Meyes, C. W. de Puiseau, A. Posada-Moreno, and T. Meisen. "Under the Hood of Neural Networks: Characterizing Learned Representations by Functional Neuron Populations and Network Ablations." In: *CoRR* abs/2004.01254 (2020). arXiv: 2004.01254.

[69] Michelle Marie Grau, Sheri Sheppard, Shannon K. Gilmartin, and Beth Rieken. "What Do You Want to Do with Your Life? Insights into how Engineering Students Think about their Future Career Plans." In: 2016.

[70] T. Mikolov, K. Chen, G. Corrado, and J. Dean. *Efficient Estimation of Word Representations in Vector Space*. 2013.

[71] R. J. Mislevy. In: *Journal of Educational Statistics* 16.2 (1991), pp. 150–155. ISSN: 03629791.

[72] C. Molnar. *Interpretable machine learning: A guide for making black box models explainable*. 1st edition. Zürich: Christoph Molnar, 2019. ISBN: 9780244768522.

[73] G. Montavon, W. Samek, and K.-R. Müller. *Methods for Interpreting and Understanding Deep Neural Networks*. 2017.

[74] A. Mordvintsev, C. Olah, and M. Tyka. *Inceptionism: Going Deeper into Neural Networks*. 2015.

[75] W. J. Murdoch, C. Singh, K. Kumbier, R. Abbasi-Asl, and B. Yu. "Definitions, methods, and applications in interpretable machine learning." In: *Proceedings of the National Academy of Sciences* 116.44 (Oct. 2019), pp. 22071–22080. ISSN: 1091-6490. DOI: 10.1073/pnas.1900654116.

[76] S. Na, Y. J. Choe, D.-H. Lee, and G. Kim. "Discovery of Natural Language Concepts in Individual Units of CNNs." In: *International Conference on Learning Representations*. 2019.

[77] R. Nanda and J. B. Sørensen. "Workplace peers and entrepreneurship." In: *Management science* 56.7 (2010), pp. 1116–1126.

[78] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. *Synthesizing the preferred inputs for neurons in neural networks via deep generator networks*. 2016.

[79] A. Nguyen, J. Yosinski, and J. Clune. *Multifaceted Feature Visualization: Uncovering the Different Types of Features Learned By Each Neuron in Deep Neural Networks*. 2016.

[80] M. Obschonka, N. Lee, A. Rodríguez-Pose, J. Eichstaedt, and T. Ebert. *Big Data, artificial intelligence and the geography of entrepreneurship in the United States*. May 2018. DOI: 10.31219/osf.io/c62tn.

[81] M. Obschonka and M. Stuetzer. "Integrating Psychological Approaches to Entrepreneurship: The Entrepreneurial Personality System (EPS)." In: *Small Business Economics* 49 (June 2017). DOI: 10.1007/s11187-016-9821-y.

[82] C. Olah, A. Mordvintsev, and L. Schubert. "Feature Visualization." In: *Distill* 2.11 (2017). ISSN: 2476-0757. DOI: 10.23915/distill.00007.

[83] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. "The Building Blocks of Interpretability." In: *Distill* 3 (Mar. 2018). DOI: 10.23915/distill.00010.

[84] J. Opitz and S. Burst. "Macro F1 and Macro F1." In: (Nov. 2019).

[85] I.-J. Park and H. Jung. "Relationships among future time perspective, career and organizational commitment, occupational self-efficacy, and turnover intention." In: *Social Behavior and Personality: an international journal* 43.9 (2015), pp. 1547–1561.

[86] S. Parrigon, S. E. Woo, L. Tay, and T. Wang. "CAPTION-ing the Situation: A Lexically-Derived Taxonomy of Psychological Situation Characteristics." In: *Journal of Personality and Social Psychology* 112 (Apr. 2017), pp. 642–681. DOI: 10.1037/pspp0000111.

[87] J. Pennebaker, R. Booth, R. Boyd, and M. Francis. "Linguistic Inquiry and Word Count: LIWC2015." In: (Sept. 2015).

[88] J. Pennebaker, M. Francis, and R. Booth. "Linguistic inquiry and word count (LIWC)." In: (Jan. 1999).

[89] J. Pennington, R. Socher, and C. Manning. "Glove: Global Vectors for Word Representation." In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Ed. by Q. C. R. I. Alessandro Moschitti, G. Bo Pang, and U. o. A. Walter Daelemans. Stroudsburg, PA, USA: Association for Computational Linguistics, pp. 1532–1543. DOI: 10.3115/v1/D14-1162.

[90] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer. *Deep contextualized word representations*. 2018.

[91] N. Poerner, B. Roth, and H. Schütze. *Evaluating neural network explanation methods using hybrid documents and morphological agreement*. 2018.

[92] N. Poerner, B. Roth, and H. Schütze. *Interpretable Textual Neuron Representations for NLP*. 2018.

[93] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever. "Language Models are Unsupervised Multitask Learners." In: 2019.

[94] S. M. Renz, J. M. Carrington, and T. A. Badger. "Two Strategies for Qualitative Content Analysis: An Intramethod Approach to Triangulation." In: *Qualitative Health Research* 28.5 (2018). PMID: 29424274, pp. 824–831. DOI: 10.1177/1049732317753586. eprint: https://doi.org/10.1177/1049732317753586.

[95] M. T. Ribeiro, S. Singh, and C. Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. 2016.

[96] M. E. Roberts, B. M. Stewart, D. Tingley, C. Lucas, J. Leder-Luis, S. Gadarian, B. Albertson, and D. G. Rand. "Structural topic models for open ended survey responses." In: *American Journal of Political Science* 58 (2014), pp. 1064–1082.

[97] S. Ruder. *NLP's ImageNet moment has arrived*. 2018.

[98] H. Sajjad, N. Kokhlikyan, F. Dalvi, and N. Durrani. "Fine-grained Interpretation and Causation Analysis in Deep NLP Models." In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorials*. Online, June 2021.

[99] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. *DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter*. 2019.

[100] M. Schar, S. Gilmartin, A. Harris, B. Rieken, and S. Sheppard. "Innovation Self-Efficacy: A Very Brief Measure for Engineering Students." In: June 2017. DOI: 10.18260/1-2--28533.

[101] M. Schar, S. Gilmartin, B. Rieken, S. Brunhaver, H. Chen, and S. Sheppard. "The Making of an Innovative Engineer: Academic and Life Experiences that Shape Engineering Task and Innovation Self-Efficacy." In: June 2017. DOI: 10.18260/1-2--28986.

[102] S. Scott and R. Khurana. "Bringing Individuals Back In: The Effects of Career Experience on New Firm Founding." In: *Industrial and Corporate Change* 12 (Feb. 2003), pp. 519–543. DOI: 10.5465/APBPP.2001.6133762.

[103] S. G. Scott and R. A. Bruce. "Determinants of Innovative Behavior: A Path Model of Individual Innovation in the Workplace." In: *The Academy of Management Journal* 37.3 (1994), pp. 580–607. ISSN: 00014273.

[104] M. Setzu, R. Guidotti, A. Monreale, F. Turini, D. Pedreschi, and F. Giannotti. "GLocalX - From Local to Global Explanations of Black Box AI Models." In: *Artificial Intelligence* 294 (2021), p. 103457. ISSN: 0004-3702. DOI: https://doi.org/10.1016/j.artint.2021.103457.

[105] M. Shanker, M. Hu, and M. Hung. "Effect of data standardization on neural network training." In: *Omega* 24.4 (1996), pp. 385–397. ISSN: 0305-0483. DOI: https://doi.org/10.1016/0305-0483(96)00010-2.

[106] L. S. Shapley. "17. A Value for n-Person Games:" in: *Contributions to the Theory of Games (AM-28), Volume II*. Ed. by H. W. Kuhn and A. W. Tucker. Princeton University Press, 2016, pp. 307–318. DOI: doi:10.1515/9781400881970-018.

[107] K. Simonyan, A. Vedaldi, and A. Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2013.

[108] R. Somasundaram and R. Nedunchezhian. "Evaluation of three simple imputation methods for enhancing preprocessing of data with missing values." In: *International Journal of Computer Applications* 21.10 (2011), pp. 14–19.

[109] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. *Striving for Simplicity: The All Convolutional Net*. 2014.

[110] P. J. Stone, R. F. Bales, J. Z. Namenwirth, and D. M. Ogilvie. "The general inquirer: A computer system for content analysis and retrieval based on the sentence as a unit of information." In: *Behavioral Science* 7.4 (1962), pp. 484–498. DOI: https://doi.org/10.1002/bs.3830070412. eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/bs.3830070412.

[111] T. E. Stuart and W. W. Ding. "When Do Scientists Become Entrepreneurs? The Social Structural Antecedents of Commercial Activity in the Academic Life Sciences." In: *American Journal of Sociology* 112.1 (2006), pp. 97–144. ISSN: 00029602, 15375390.

[112] M. Sundararajan, K. Dhamdhere, and A. Agarwal. "The Shapley Taylor Interaction Index." In: *ICML*. 2020.

[113] M. Sundararajan and A. Najmi. *The many Shapley values for model explanation*.

[114] M. Sundararajan, A. Taly, and Q. Yan. *Axiomatic Attribution for Deep Networks*. 2017.

[115] C. Tantithamthavorn, A. E. Hassan, and K. Matsumoto. "The Impact of Class Rebalancing Techniques on the Performance and Interpretation of Defect Prediction Models." In: *IEEE Transactions on Software Engineering* 46.11 (2020), pp. 1200–1219. DOI: 10.1109/TSE.2018.2876537.

[116] E. Tjoa and C. Guan. *A Survey on Explainable Artificial Intelligence (XAI): Towards Medical XAI*. 2019.

[117] L. Torroba Hennigen, A. Williams, and R. Cotterell. "Intrinsic Probing through Dimension Selection." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 197–216. DOI: 10.18653/v1/2020.emnlp-main.15.

[118] M. Tsang, S. Rambhatla, and Y. Liu. *How does this interaction affect me? Interpretable attribution for feature interactions*. 2020.

[119] M. Valipour, E.-S. Lee, J. Jamacaro, and C. Bessega. "Unsupervised Transfer Learning via BERT Neuron Selection." In: (Dec. 2019).

[120] J. J. Vaske. *Survey research and analysis*. ERIC, 2019.

[121] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. *Attention Is All You Need*. 2017.

[122] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush. "Transformers: State-of-the-Art Natural Language Processing." In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Online: Association for Computational Linguistics, Oct. 2020, pp. 38–45.

[123] C.-K. Yeh, B. Kim, S. Arik, C.-L. Li, T. Pfister, and P. Ravikumar. "On Completeness-aware Concept-Based Explanations in Deep Neural Networks." In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 20554–20565.

[124] M. D. Zeiler and R. Fergus. *Visualizing and Understanding Convolutional Networks*. 2013.

[125] Y. Zhang, R. Jin, and Z.-H. Zhou. "Understanding bag-of-words model: A statistical framework." In: *International Journal of Machine Learning and Cybernetics* 1 (Dec. 2010), pp. 43–52. DOI: 10.1007/s13042-010-0001-0.