HOGESCHOOL UTRECHT

Master of Informatics

Automatic classification of short-answers for semi-structured open-ended questions with only a small dataset: A Master thesis

Stijn Smulders

HU University of Applied Sciences
P.O. box 182
3500 AD UTRECHT
The Netherlands

Supervisor: Philippine Waisvisz

Date: 01/03/2023

***Abstract**: The recent advancements in machine learning (ML) techniques have significantly improved the performance in various natural language processing (NLP) tasks, particularly in deep learning with the transformer architecture. Despite their promising applications in different domains, these innovations have yet to be fully utilized in the education sector. This can be attributed to the large amount of data required for these ML techniques, which schools may not have access to. This thesis aims to address this gap by answering the following question: How can semi-structured open-ended questions in student homework assignments automatically be classified by a model that is trained on a small dataset (n=~100) using modern ML techniques? A literature review of commonly used ML techniques is conducted, followed by the development of a prototype for a specific case study. The results demonstrate the viability of automatic classification of homework assignments, although performance is suboptimal for certain types of questions. A RobBERTa-like model is employed for classifying 7 different types of semi-open questions, with promising results observed for 'less complex questions'. The results suggest a correlation between question complexity and classification accuracy. A set of guidelines is proposed to aid future developers in applying similar techniques to their own datasets.*

Acknowledgment

During the last 2.5 years, I’ve learned a lot about the world of Data Science, research, and working with machine learning models. Specifically on the topic of machine learning and natural language processing. Exploring this domain has been enormously educational but I could not have done it without the help of so many people. I want to thank everyone who supported me in this process of learning and discovering and contributed in any way to my research.

First of all, I want to thank Philippine Waisvisz for helping me navigate the world of scientific research and helping me with structuring my work and thesis.

Next to Philippine, I want to thank Sietske Tacoma who helped me get started with my research. Her guidance during these first few months helped me in scoping and formulating my research questions.

I want to thank everyone who participated in the final focus group to discuss the developed prototypes. The input from that focus group allowed me to take the final steps in my research.

Finally, I would like to thank my wife Robin Smulders for all the support during the last 2.5 years.

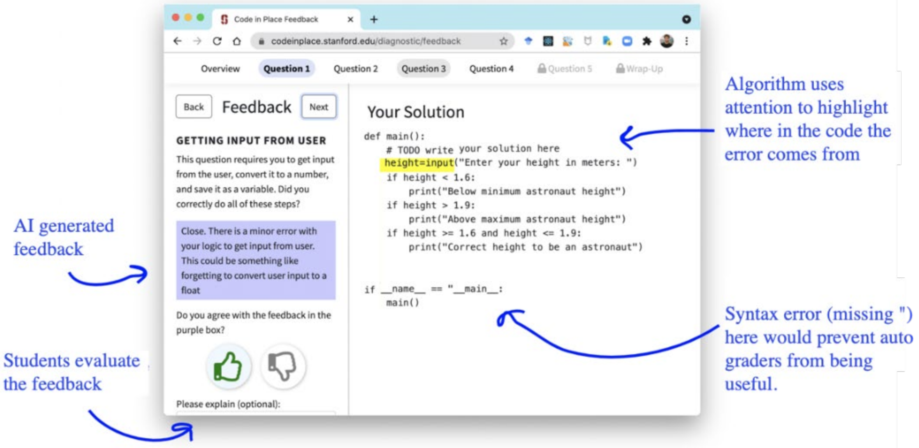
Table of Contents

Contents

1. Introduction

In higher education, students receive homework assignments to practice new skills or test their knowledge. Commonly, students receive global feedback on their work by discussing different solutions in the classroom or by being able to compare their solutions to that of the teacher, it is not often that students receive specific feedback on their own work. This is because providing custom feedback on homework assignments is a time-consuming process. Often teachers choose formats like multiple-choice questions because they can be checked automatically and leave out open-ended questions even though open-ended questions are considered to be a better fit to test skills such as critical thinking and synthesis (Polat, 2020). Considering the role of feedback and its importance in learning new skills (Ferguson, 2009), students are missing out on an important part of learning.

In 2021 the University of Stanford tried to solve this problem with machine learning (ML) (Wu et al., 2021). They created an application that could automatically classify student programming assignments and provide feedback (using modern machine-learning techniques) (image 1).



GETTING INPUT FROM USER

This question requires you to get input from the user, convert it to a number, and save it as a variable. Did you correctly do all of these steps?

Close. There is a minor error with your logic to get input from user. This could be something like forgetting to convert user input to a float

Do you agree with the feedback in the purple box?

Please explain (optional):

```
def main():
    # TODO write your solution here
    height=input("Enter your height in meters: ")
    if height < 1.6:
        print("Below minimum astronaut height")
    if height > 1.9:
        print("Above maximum astronaut height")
    if height >= 1.6 and height <= 1.9:
        print("Correct height to be an astronaut")


if __name__ == "__main__:
    main()
```

AI generated feedback

Students evaluate the feedback

Algorithm uses attention to highlight where in the code the error comes from

Syntax error (missing ") here would prevent auto graders from being useful.

Image 1. Meta-Learning Student Feedback to 16,000 Solutions (Mike Wu et al., 2021)

This model did not only automatically grade the student homework assignment but also applied multi-classification to decide what parts of a rubric (rating model) applied. This created the possibility to provide a student with more specific feedback on their work than just black-box testing. This model was deployed in a real-life setting with 12,000 students with great success.

> *"Across 16,000 pieces of feedback given, students agreed with the AI feedback **97.9%** of the time, compared to 96.7% agreement to feedback provided by human instructors. " (Mike Wu et al., 2021)*

This shows the potential that machine learning has in solving this feedback problem, but not a lot of colleges or universities are currently using the full potential of the capabilities that AI has to offer.

1.1 Problem statement

Machine learning models often require large quantities of data to be trained to achieve desirable performance. Not all universities or colleges have access to these vast amounts of data to train these types of ML models. With the rise of new modern ML techniques, such as the new Transformer architecture (Vaswani et al., 2017), powerful models have arisen that are pre-trained (unsupervised) on vast amounts of data and do not require as much data to achieve high-end results on specific tasks (fine-tuning). But the field of Machine learning moves fast and these solutions have not found their way to the classroom yet.

As stated previously, feedback is an important part of learning (Ferguson, 2009) but students often receive feedback at the end of a module when taking the final test to prove their competence. Manually giving feedback on homework assignments could solve this issue but this is a time-consuming process for the teacher that does not scale with larger groups of students.

Modern ML techniques could help solve this issue but no copy-paste solution is available to automatically classify student homework. An exploration of the field and a set of guidelines for developers could be a good first step to tackling this issue.

1.2 Research Question

The goal of this research is to find out if it is possible to classify student homework assignments if you only have a small dataset. This results in the development of a set of guidelines that can help with choosing the right techniques to support the automatic classification of student homework. At the end of the thesis, the following questions shall be answered: *How can semi-structured* open-ended *questions in student homework assignments automatically be classified by a model that is trained on a small dataset (n=~100) using modern ML techniques as to provide students with formative feedback without costing more time?*

To answer this question, the following questions need to be answered first:

1. *What kind of techniques are currently used to classify text?*
2. *What are the properties of homework assignments from a data perspective?*
3. *What type of machine learning model is well suited to classify homework assignments?*

The goal is not to find the absolute best way to classify homework assignments but to explore the field of machine learning in the context of the domain of education (figure 1) and find effective ways to automatically classify open-ended questions in student homework assignments. More information on how these questions are answered and the chosen methodology can be found in chapter 4.

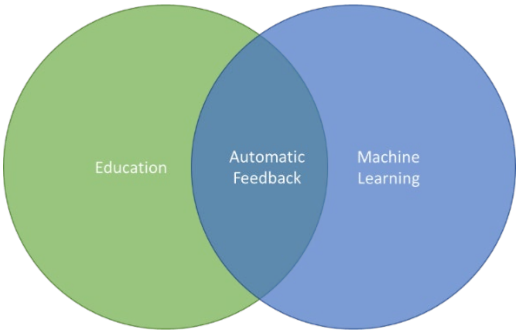


Figure 1: Venn diagram domain and field overlap.

1.3 Scientific and practical contribution

Automating the process of giving feedback would help students to determine what subjects need more attention without increasing the time required by teachers to review this work by hand. Not only will automating this feedback process reduce the time needed to review student homework, but it will also allow students to receive feedback on demand (because it is automated) which may improve the usefulness of feedback (Shute, 2008).

The exploration of related work shows the potential of using statistic models or machine learning models to automate this process but most examples are based on large datasets like that from massive online open courses (Wu et al., 2021). Most courses within higher education only have a few hundred students attending which results in much smaller datasets.

Previous research shows the exploration of automatically supplying student homework assignments with feedback (or automatically grading student homework) (L. Zhang et al., 2019) (X. Wang et al., 2020) (Alhami, 2011) but these papers do not use modern ML techniques such as Transformers (some even recommend future research on this topic by testing out these modern techniques).

A set of guidelines could help future developers with automating the process of giving feedback on student homework to increase the quality of education. Not only will automating this process give students the ability to receive feedback on homework assignments but it will also create the possibility to access the feedback on-demand and as often as they want. This can increase the autonomy of the students which plays a big role in implementing blended learning (Kintu et al., 2017) which is currently a popular teaching approach for many universities.

2. Theoretical Background

A review of relevant literature has been conducted to underline the importance of providing feedback, find examples of applying machine learning models to deal with text classification, and research on how to train these models with only a small dataset. For papers on the subject of feedback, multiple famous works (many citations) were summarized to give a better understanding of the importance of feedback This explains why automatically creating feedback could improve student learning and the difference between providing feedback on open questions and multiple-choice questions.

On the subject of text classification, a total of 24 papers were reviewed containing examples of text classification. Some in the domain of student homework assignments and some in other domains. The most important term that was used to find these papers were 'automatic feedback' or 'text classification' and they were selected on their date of publishing (after 2018, the year the new ML architecture Transformers was published). After finding these papers a deep dive followed into the most used ML techniques such as working with BERT (Bidirectional Encoder Representations from Transformers).

In chapter 5.1 a summary of common techniques can be found describing common ML techniques (in the domain of text classification) and a reference to the used papers for this list can be found in Appendix B.

2.1 Importance of feedback

We've known the importance of feedback for a while now (Hattie & Timperley, 2016), but not often do students receive feedback when they need it most (while learning and practicing new subjects). Previous studies show the importance of adequately timed feedback (Shute, 2008), but commonly students will study a subject for several weeks and in the end, receive a test or an exam (summative assessment) that is graded to decide if they failed or passed the subject with not many feedback moments in-between.

Some teachers will use formative assessments during the course so that students can test their knowledge or skills. These types of formative assessments are considered an important mechanism for improving student learning (Gedye, 2010). As Shute states (Shute, 2008) formative feedback is usually presented as information to learning in response to some action, in this case answering a question. In the case of open-ended questions, a student should get the information if the answer is correct (response accuracy) or hints why it is not correct. Formative feedback should be non-evaluative, supportive, timely, and specific (Shute, 2008).

Open-ended questions and multiple-choice questions appear equivalent in their ability to assess knowledge of a subject (Hift, 2014), but when testing certain skills such as critical thinking, open-ended questions seem a better fit (Polat, 2020). Unfortunately, open-ended questions (or short answers) have to be manually checked. This is often tedious and time-consuming work. This could be a reason why teachers commonly don't choose to incorporate these types of questions into homework or formative assessments. On the other hand, these types of assessments (open-ended questions) are a popular way to summative assess a student at the end of a course. If there was a way to reduce the time needed to check assignments that contain short-answers, teachers might choose this form of formative assessment more often.

2.2 Deep learning and data

In the last decade, the field of AI has grown immensely, and specifically the field of deep learning has had a big impact on how problems are solved. One thing these deep learning solutions have in common is that they need large quantities of data to be trained. This data is needed because a neural network needs a lot of examples to figure out what features are important and what features can be ignored. Some strategies allow working with ML models even with a smaller dataset (Y. Zhang & Ling, 2018) and previous research is available that explores the performance of ML models with different sizes of datasets (Brigato & Iocchi, 2020).

2.3 Attention is all you need

New machine-learning techniques are discovered almost daily and these discoveries are the drive behind innovation and new solutions to existing problems. In 2017 a new ML architecture called Transformer was proposed. This architecture revolutionized the natural language processing world and transformer-based models started to pop up everywhere (Gillioz et al., 2020).

Based on this Transformer architecture Google developed BERT (Devlin et al., 2018a) which outperforms traditional machine learning techniques in common natural language processing (NLP) tasks and has become the default technique for NLP problems (González-Carvajal & Garrido-Merchán, 2020).

Although BERT has become a default for common NLP tasks most studies utilize large datasets to train models and to achieve high accuracy. Specifically, when working with a smaller dataset, more traditional ML models such as LSTM (Long Short-Term Memory) can still outperform BERT (Ezen-Can, 2020) (Jiao et al., 2020), although other research shows the opposite where Transformer-based models show great results when trained on just a small data set (Brown et al., 2020).

2.4 Related experiments

Specifically in the domain of education previous work is available to learn from. Some of this previous work has been conducted before 2017 (release of the transformer architecture).

Some experiments from a few years back used a simple word count (bag of words) to compare if a student's answer could be similar to the expected answer (Alhami, 2011). But with the rise of more complex neural networks, new solutions emerged. Stanford University showed that it was possible to add automized feedback to submitted programming assignments (Wu et al., 2021). Students participating in this experiment indicated that they could not distinguish between the feedback of a teacher assistant and that of the model (Wu et al., 2021). Stanford is not the only one to prove that it is possible to classify open-ended questions. The FernUniversität in Hagen experimented with labeling student test answers and providing them with recommendations while trying out different models such as a Multinomial NB, random forests, and more (X. Wang et al., 2020). Or you can look at the experiment from Zhang et al., 2019, where they used a transformer to embed the text and compared this to an LSTM-based model to categorize the short text answers. All these experiments show that it is possible to categorize text with (different) machine learning models. Next to categorization, there are more possibilities within machine learning such as next-sentence prediction. Experiments like predicting code (a form of semi-structured text) summaries (Allamanis & Sutton, 2016) show that machine learning models are capable of finding patterns within the text.

What these experiments have in common is the large amounts of data they require to get a model with high performance. As stated previously teachers do not always have access to these large datasets. There have been experiments where different machine learning models are compared on different-sized datasets (Vabalas et al., 2019) or comparing models on different types of (small) datasets to see what type of models perform the best (Antoniou & Storkey, 2018). These experiments show that even if the dataset is small, it is possible to develop an accurate classification model. If we specifically look at classifying semi-structured text there is a step called word embedding necessary (because the input of a model needs to be a number, not a string). An experiment at Cardiff University (Allamanis et al., 2018) shows the difference between using a custom model and using a pre-trained model for embedding such as BERT (Devlin et al., 2018). In their experiment, they try out different sizes of datasets ranging from 200 to 5,000 observations. While pre-trained models like BERT show promising results, they do not always outperform simple custom-built models for word embedding. Whilst an experiment with patient messages shows the potential of pre-trained models (Si et al., 2020), other experiments (Kumar Sharma et al., 2018) show that training your own machine-learning model is also viable.

When a model was developed using BERT, papers often addressed the difference between domain-specific text and domain-general texts (Si et al., 2020). An example of domain-specific text could be a case document that students need to read to answer the questions and an example of domain-general text could be pages on Wikipedia (not related to a specific subject). Another aspect that can be relevant to the performance of a model is the diversity within the data (variety) (Edwards et al., 2020). Because most pre-trained models are based on an English or international vocabulary the language of the data can be relevant for model performance (de Vries et al., 2019).

There are some discrepancies between the use of deep learning methods such as neural networks and more traditional methods such as random forests and just using a simple bag of words technique (Antoniou & Storkey, 2018). When using pre-trained models like BERT the size of an observation (word count) is relevant because these models only accept a maximum of 512 tokens (words).

All in all the selected papers show the potential of categorizing student homework even with small quantities of data but there are many discrepancies on how to achieve this goal. There is no clear view of what specific technique should be used. In chapter 5.1 commonly used techniques in previous work are summarized and further explained.

A side note on this potential is the methodology chosen for the research in those papers. Most of the selected papers describe an experiment with a specific dataset and a model that performs well on that dataset. It is hard to establish if the findings of these experiments generalize well enough to be applied to other datasets or domains.

3. Case Study

A dataset is available containing labeled data of 173 students on an open-question test for a course on technical business. This test consists of 15 open-question questions that can be answered by reading a case document. Here are some of the characteristics of this exam:

- The questions, answers, and case document are in Dutch;
  - the students that took this exam are first-year students of a technical business university (Bachelor) with an age of around 20 years old.
- The exam has been taken in an online exam environment called Remindo.
- The case document contains 703 words.

Question: Noem een voorbeeld van een beheerscyclus.
Answer: Plan-do-check-act cyclus

Example - Example of one of the questions

This dataset was chosen because it contains different types of open questions with different types of complexity. It will be used to get a better understanding of the definition of student homework and to develop a prototype to evaluate the performance of the different types of machine learning models.

This dataset is well suited to solve the feedback problem because the answers are scored via a Rubric that contains specific criteria that need to be met to get that score. This means that every score is linked to a set of criteria that the student sees when the result of the test is published.

A full list of all the questions is available in Appendix A. The answers given by students are not published in this study because of the regulations of the client.

This dataset was first exported from an online test environment called Remindo in one large JSON file and later transformed into a simple CSV containing all the answers to all the questions. This CSV has the following columns:

- **qid** - Question ID
- **answer** – Text containing the student's answer.
- **score** – Points that are given by a teacher.
- **max_score** – Maximum number of points that could be given for this question.

Further details on this specific dataset are explored in chapter 5.2.

4. Research Method

"Pragmatism is a school of thought that considers practical consequences or real effects to be vital components of both meaning and truth."
~ Hevner, 2007

This research is designed by following the design science methodology of Hevner ((A. R. Hevner, 2007). First, the choice of methodology is explained followed by a list of research methods that have been used to conduct this research.

4.1 Chosen Methodology

The goal of this research is to explore the field of ML in the context of education and specifically on the subject of automatic feedback. This will be achieved by looking at the current knowledge base and finding examples of good solutions and applying these solutions in a new domain where they have not been tested yet. The scientific contribution will be any extensions to the original theories and methods made during the research (A. Hevner, 2004). These findings will be summarized in the form of guidelines.

The method to achieve this goal is based on the design science research cycles by Hevner (A. R. Hevner, 2007) (Figure 2). Because of the scope of this research and the time available, only a few iterations will be executed. Any unanswered questions that arise during the research are described in chapter 6.3, Discussion, and chapter 6.4, Future works.

This method was chosen because, during the literature review for the theoretical background, many papers on the subject of applying machine learning models used a simial approach. Common methods in design science research (as can be seen in figure 2) are applied in those papers to create artifacts and evaluate these artifacts.

This method is well aligned with an objective ontology with a positive paradigm. (O'gorman & Macintosh, 2015). An objective ontology in the sense that the research question can be answered by observing behaviors and testing out previous findings in a new context, and positive in the sense that there is a focus on facts and hypotheses will be formulated and tested.

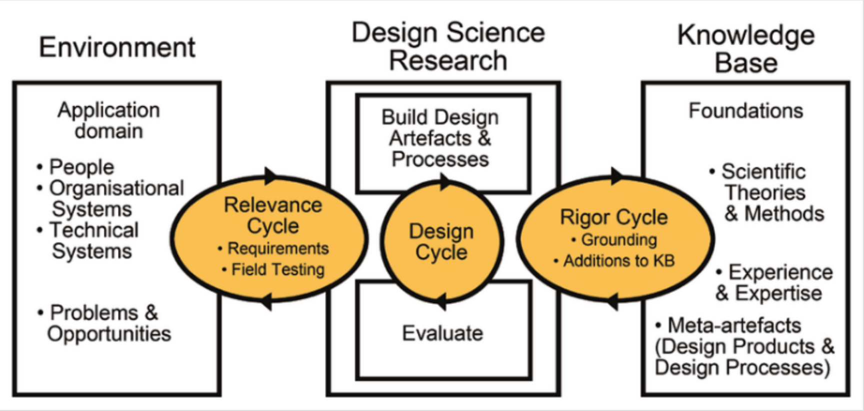


Figure 2. Design Science Research Cycles (A. R. Hevner, 2007)

Based on this methodology the following research methods have been chosen to answer the sub-questions of this research (chapter 1.2). These methods have been placed within the diagram of the methodology (figure 3) and explained further down this chapter.

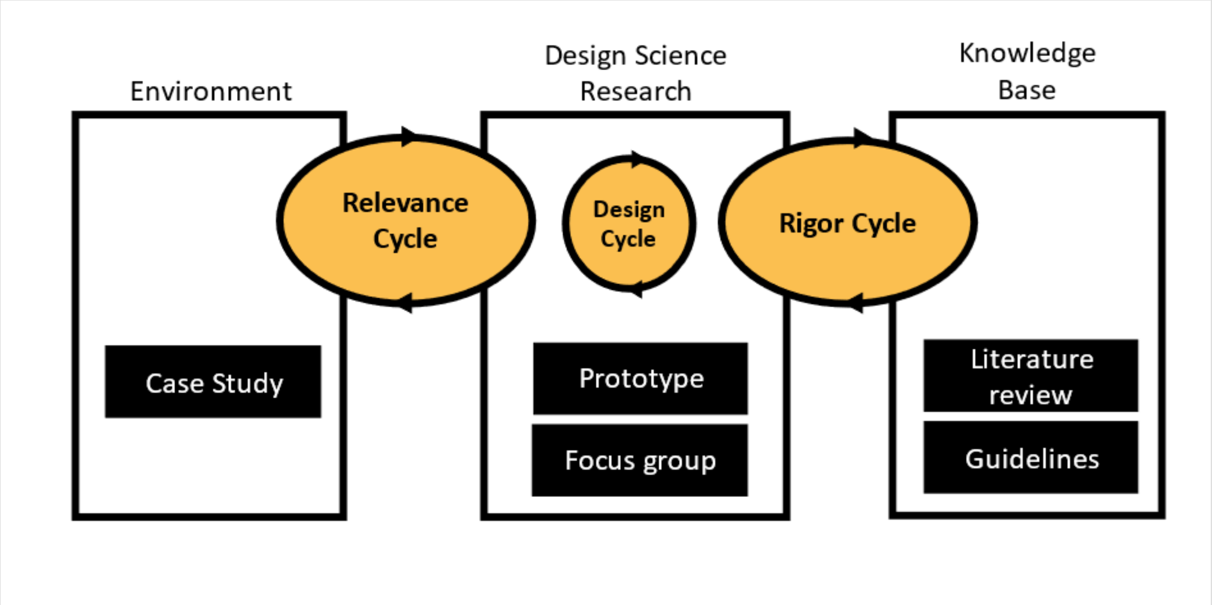


Figure 3. Specific methods distributed by methodology.

4.2 Literature Review

The goal is not to design a new model or solution to classify student answers but to evaluate best practices and apply them to the domain of education. During the literature review, these solutions will be identified and inventoried. This review will specifically focus on experiments and previous work that covers the subject of text classification (not limited to homework assignments).

The result is an overview of available techniques and solutions that answers sub-question 1 and is used as the input to develop the prototype.

4.3 Case Study

With this method, the available dataset is analyzed (statistically) and its properties are charted to give a better understanding of the complexity of the data. Specifically on the subject of structured versus semi-structured. Next to this quantitative analysis, a more qualitative analysis of the dataset is accomplished by interviewing different stakeholders on the subject of classifying text to get a better view of the possibilities and to determine what types of classification will be easy and what types will be hard.

The result is a definition of the complexity of the questions that can help to choose the right techniques for the development of the prototype.

4.4 Development of a prototype

Based on the literature review and data analysis a prototype will be developed and evaluated to test the performance of different techniques on different types of questions. More information about the prototypes can be found in chapter 5.3.

The performance of the model is not the sole indicator to determine the success of the research paper as different solutions can be optimal for different situations (datasets). The prototype is only a tool to help during the evaluation and as input to formulate guidelines.

As a result of this method, sub-questions 3 and 4 will be answered and all findings will be input for the final evaluation.

4.5 Focus group

The focus group was organized to evaluate if the chosen techniques are applied correctly. The final results are evaluated in different ways. The results are discussed with domain experts and by forming a focus group on the subject of text classification to evaluate the developed prototype. During this focus group, the subjects of working with a small data set, data set imbalance, and working with BERT are discussed. Because of the scope of this project, some of the results are formulated as points of discussion (chapter 6.3), and some results are used to formulate guidelines (chapter 6.2).

4.6 Guidelines

As a result of this method, a set of guidelines (lessons learned) are formulated. These guidelines are formed using the results from the different research methods and grouped by the different categories of the design science research cycle. The guidelines are formulated based on the results of the different research questions and grouped by the research methodology.

4.7 Validity

The validity of this project is guaranteed by creating reproducible results (as far as possible). All steps taken and techniques chosen are described in this Thesis allowing other people to replicate these steps and see if the results are reproducible. Unfortionaly only one dataset is used within the case study which makes it harder to conclude if the results are transferable.

In chapter 6.3 any threats to this validity are presented and grouped into the following categories: Construction, Internal and External (Wohlin et al., 2000).

5. Findings

The results of this research are ordered by the research questions. For every question, a partial conclusion is drawn and the final conclusion is described in chapter 7. Next to this partial conclusion, some findings are formulated as hypotheses that are answered further down in this chapter.

The first results show an overview of commonly used ML techniques (based on a literature review). Next to this literature review a data analysis was performed on the dataset from the case study. Based on these results a prototype was built and evaluated.

5.1 What techniques are currently used to classify text?

This question was answered with a literature review on examples of text classification to find out what types of models or techniques are commonly used. Twelve papers were selected that are released in the last 4 years and popular techniques were inventoried. A full list of these papers can be found in Appendix B. These papers were found via Google Scholar when searching with the terms ‘text classification’ and selected based on checking the abstract and finding papers that test specific model types against certain baseline models. The most common techniques and methods are explained to give a better understanding of how these techniques work and to find out what techniques will be used in the development of a prototype (research question 3, chapter 5.3).

Text classification in a nutshell

To classify text there are always some steps that need to be taken to before a model can automatically predict a label or category. An important step is turning raw text into features (figure 4). Because statistical models cannot work with words directly, these words need to be transformed into some sort of numbers (embedding).

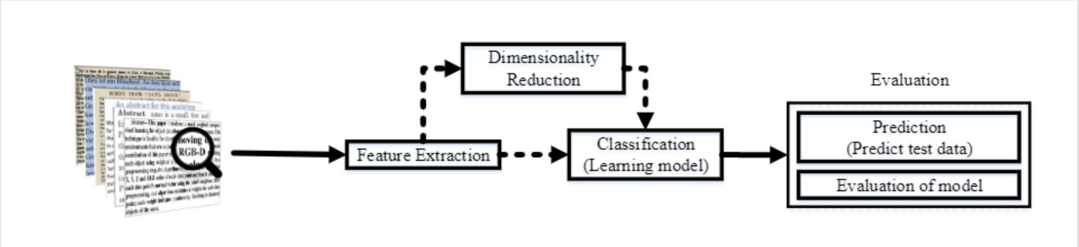


Figure 4: Text classification pipeline (Kowsari et al., 2019)

In more traditional text classification approaches this step of feature extraction is commonly achieved by tokenizing the text Tokenizing means breaking a stream of words into meaningful elements called tokens. It is important to solve common natural language processing issues such as stop words, abbreviations, slang, and/or spelling corrections (cleaning). But in more modern text classification approaches, such as models based on the Transformer architecture (Vaswani et al., 2017), this step of feature extraction and text cleaning is part of the given solution.

The following chapters show examples for each of these steps and possible techniques that can be used during these steps.

5.1.1 Common embedding techniques.

In the more traditional text classification method, an important step is using word embeddings to extract features from the text. This is needed because machine learning models cannot accept the text as input but require numbers. This feature extraction step can be seen as turning text into meaningful numbers. These are often vectors that contain dimensional data about a word or a sentence. After this feature extraction, a classification model can be applied. A review of word embedding techniques (Selva Birunda & Kanniga Devi, 2021) shows the most common word embedding techniques (figure 5) placed into 3 categories:

1. Traditional Word Embedding.
2. Static Word Embedding.
3. Contextualized Word Embedding.

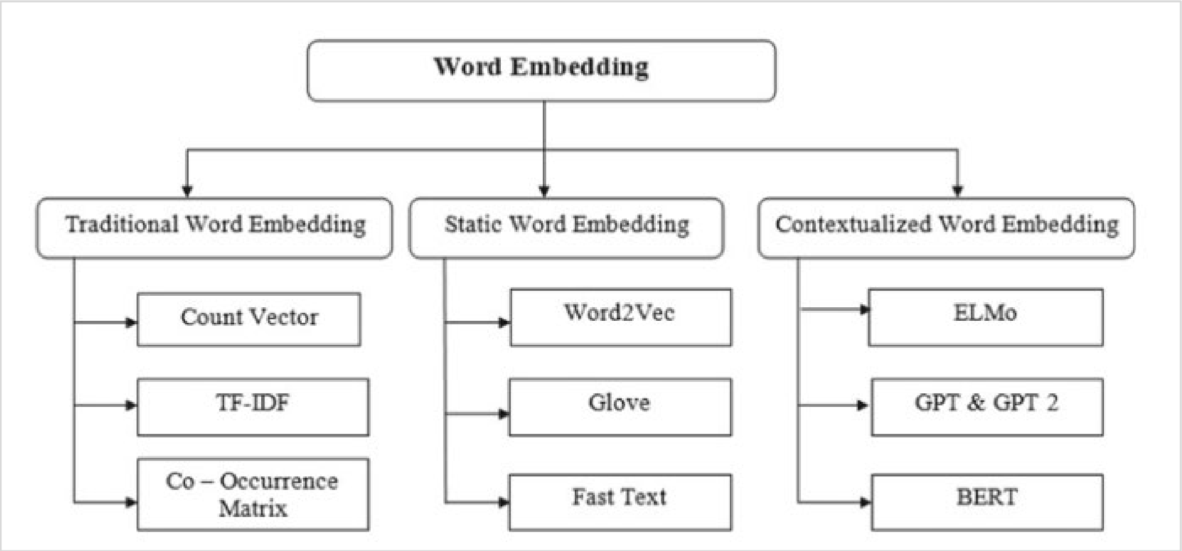


Figure 5. Type of word embeddings. (Selva Birunda & Kanniga Devi, 2021)

Traditional Word Embeddings

The most traditional word embeddings use statistical techniques to show the significance of a word in a document but they have no awareness of the context of a word. For example, the word ‘my mouse broke down’ could be about the mouse connected to my computer or a living animal. These types of models do not have a good way to deal with this context or this word sense disambiguation.

A solution to fix this issue with context is to add n-grams to your embedding. Combining multiple words gives a better understanding of the context the word is in but this does not solve the issue of context entirely (for example if there are many words in-between that influence context).

Sentence: “The quick brown fox”
Normal Embedding: [The, Quick, Brown, Fox]
Embedding with n-grams of 2: [The, Quick, Brown, Fox, The Quick, Quick Brown, Brown Fox].

Example - N-gram of 2.

Static Word Embeddings

Static Word Embeddings have the same issue with context. These models are often pre-trained on a large dataset but although they are better at finding context within a sentence (what words are often seen next to each other), they still struggle to find context in-between sentences.

Contextualized Word Embeddings

The final category falls into the group of NLP with transformers. These word embeddings output vectors that can vary based on the context of the sentence in which the word appears.

5.1.2 Traditional text classification techniques

There are too many classification techniques available to list in this thesis. In the field of text classification, some techniques are used more often than others. Previous studies have been conducted to compare different traditional text classification models to deep learning models (González-Carvajal & Garrido-Merchán, 2020) (Zulqarnain et al., 2020) (Kowsari et al., 2019). From these papers the following 3 techniques are used most commonly:

Multinomial Naïve Bayes: One of the easiest classification models out there. It calculates the probability that a feature (in the case of text a word or an n-gram of words) is correlated to the labeled class.

Random Forest: An older model based on multiple decision trees (bagging). The nodes in the decision tree are based on the features in the dataset

Support Vector Machine: A robust prediction method based on statistical learning of the dataset. It can separate a dataset into multiple classes (linear by default and non-linear by using the kernel trick).

5.1.3 Modern text classification with transformers.

In 2017 a paper was published called “Attention is all you need” (Vaswani et al., 2017), which introduced the transformer architecture (image 2). This architecture solved some problems with the then-popular architectures such as LSTM (Long Short-Term Memory) and RNN (Recurrent neural networks).

This architecture works in 2 steps. Encoding and decoding. Both steps require some form of embedding to occur before they can process the data. In this step, the sentence is turned into a vector with numbers.

Text input: “The Quick Brown Fox”
Output after embedding: [0, 2740, 12573, 16990, 17673, 2]

Example - Embedding with Transformers

These numbers correspond to a token with a certain position in an embedding space. What position a token gets is determined by how the model was pre-trained.

The tokens are the input for the encoder that turns these tokens into vectors. These vectors are input for a decoder to turn the information into something useful. A common use case for these types of models is translation. The decoder will require the output that the model will need to predict and the vectors that came from the encoder. Based on this information it will try to predict the right output by turning the calculated vectors back into tokens, and those tokens back into words (the same way the embedding turned the words into tokens).

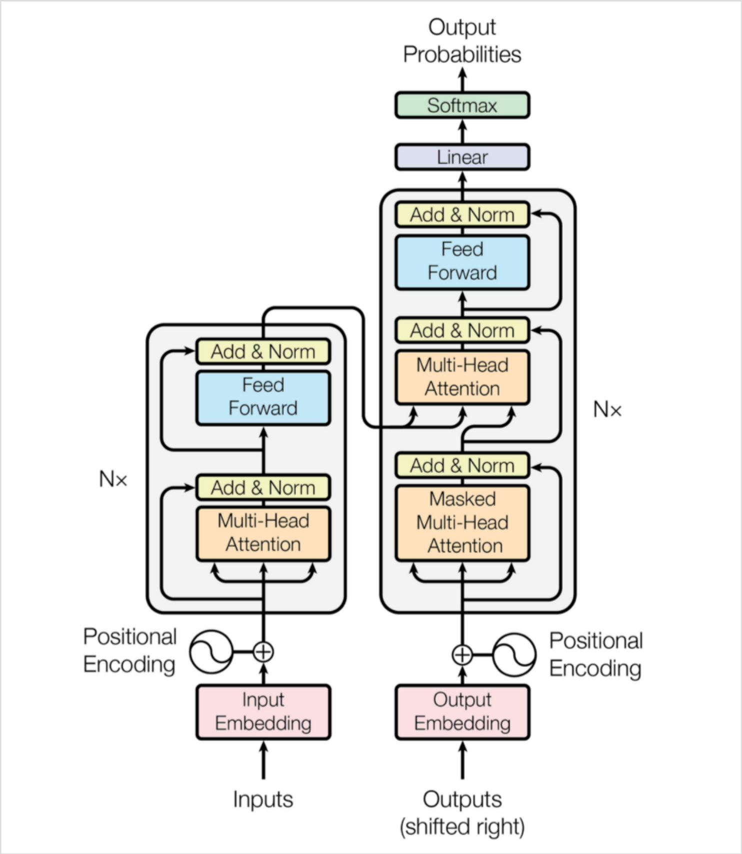


Image 2: Transformer architecture (Vaswani et al., 2017)

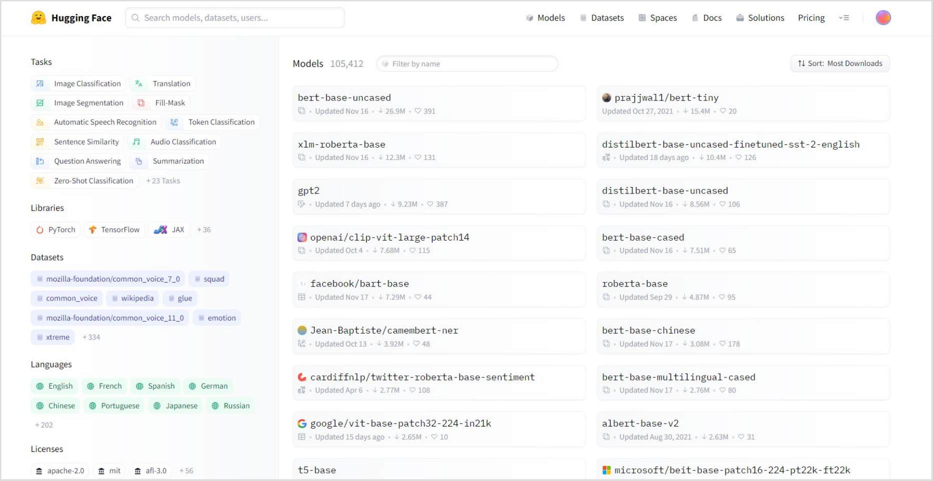
Problems with context

In many language tasks, context plays an important role. The meaning of a word is determined by its position within a sentence and the context in which it is used. RNN tried to solve this issue by looking at the entire sequence of words but this created a problem called ‘vanishing gradients’. Like most neural networks RNNs learn through backpropagation. The vanishing gradient means that this backpropagation failed to change the weights of the model to properly learn. LSTM tried to solve this problem by using short- and long-term memory. This means that it is better in understanding what information is important and what information can be forgotten, which in turn reduced the noise and problems with this vanishing gradient. But training an LSTM takes a long time because of its size and has issues with the context in longer sentences. The transformer architecture solved this issue of context by introducing ‘self-attention’ that gives a similarity score between words in a sentence and by adding a positional embedding to keep track of the position of a token in a sentence. Because of these changes, it was possible to process the sentence as a whole (non-sequential) which made it much faster to train than an LSTM and allowed the architecture to retain the context of a word even with longer sentences.

Based on this architecture new models were created that have achieved significant performance increases on common NLP tasks. One of the most famous transformer-based models is BERT (Devlin et al., 2018a) and GPT (Generative Pre-trained Transformer) (Radford et al., 2019).

BERT-based models are pre-trained using unsupervised machine learning methods and are a great starting point for building (fine-tuning) your solution. BERT focuses more on classic NLP tasks such as next-sentence prediction, masked language modeling, and text classification, whereas GPT-2 is more optimized for text generation.

For both architectures, there are a lot of starting models available on Hugging Face (image 3). BERT has a few more (around 13,000) than GPT-2 (around 7,000). Specifically, when working with the Dutch language BERT has a few models that are pre-trained on multi-language data (Roberta) (Liu et al., 2019) and even specifically Dutch data (BERTje) (de Vries et al., 2019) (RoBERTje) (Delobelle et al., 2021).



bert-base-uncased · Updated Nov 16 · 26.9M · 391

xlm-roberta-base · Updated Nov 16 · 12.3M · 131

gpt2 · Updated 7 days ago · 9.23M · 387

openai/clip-vit-large-patch14 · Updated Oct 4 · 7.68M · 115

facebook/bart-base · Updated Nov 17 · 7.29M · 44

Jean-Baptiste/camembert-ner · Updated Oct 13 · 3.92M · 48

cardiffnlp/twitter-roberta-base-sentiment · Updated Apr 6 · 2.77M · 108

google/vit-base-patch32-224-in21k · Updated 15 days ago · 2.65M · 10

t5-base

prajjwal1/bert-tiny · Updated Oct 27, 2021 · 15.4M · 20

distilbert-base-uncased-finetuned-sst-2-english · Updated 18 days ago · 10.4M · 126

distilbert-base-uncased · Updated Nov 16 · 8.56M · 106

bert-base-cased · Updated Nov 16 · 7.51M · 65

roberta-base · Updated Sep 29 · 4.87M · 95

bert-base-chinese · Updated Nov 17 · 3.08M · 178

bert-base-multilingual-cased · Updated Nov 17 · 2.76M · 80

albert-base-v2 · Updated Aug 30, 2021 · 2.63M · 31

microsoft/beit-base-patch16-224-pt22k-ft22k

Image 3 - Hugging Face, a library full of pre-trained ML models.

It is possible to train a model like BERT from scratch but it’s faster to take a pre-trained model (transfer learning) and fine-tune it using domain adaptation. This can be done by taking a pre-trained model and continuing training on a specific corpus (Bilal & Almazroi, 2022).

5.1.4 Fine tuning and Few-Shot Learning

Most neural networks need a lot of data to be trained, but when working with a Transformer based model it is recommended to use pre-trained models (transfer learning). These models have been trained unsupervised on large quantities of text. This pre-training step gives the model a better understanding of the corpus. After this pre-training step, a new model can be created by reusing the weights (transfer-learning) and then finetuned for a specific task. This pre-training step allows for the possibility of Few-

Shot Learning (FSL). Few-Shot Learning enables learning a new task with only a few examples (supervised learning) by incorporating prior knowledge (transfer learning) (Y. Wang et al., 2020). Transformer models are well suited for Few Shot Learning as the models you use are already pre-trained. In 2020 OpenAI released GPT-3 and even named the paper 'Language models are Few-Shot learners' (Brown et al., 2020). They showed that large language models such as GPT-3 can learn to perform a wide range of natural language processing tasks with little task-specific training data and they demonstrate that these large models increase the performance of few-shot learning. Image 4 shows how over the last few years these models have grown in size. In this image, it becomes clear the model GPT-3 has more than 500 times the parameters than the largest BERT model.

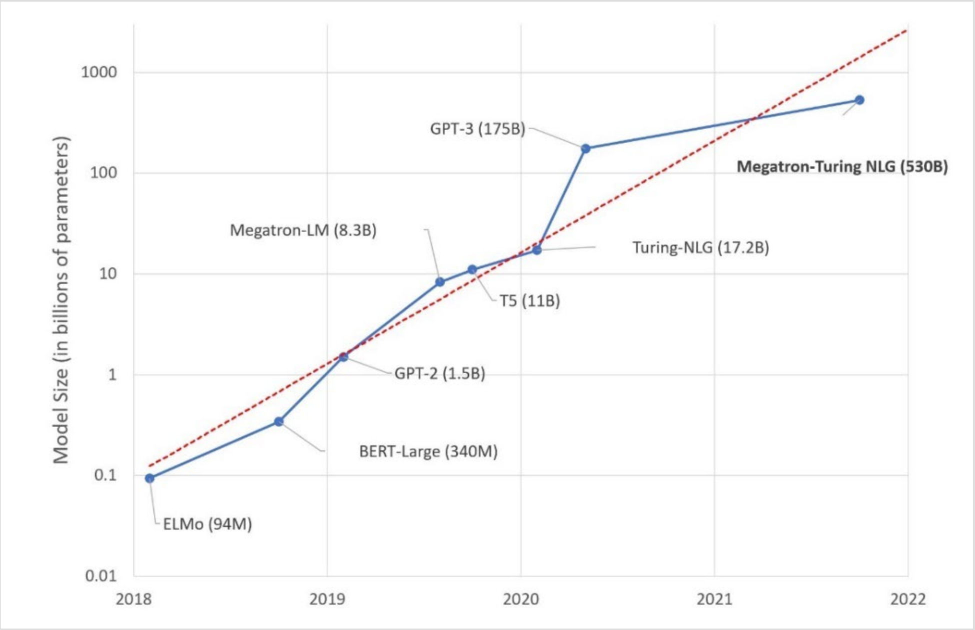


Image 4 – Models with their size as the number of parameters (Ali Alvi & Paresh Kharya, 2021)

5.1.5 Partial conclusion

Machine learning techniques for text classification can be grouped into two categories: Traditional ML techniques and modern machine learning techniques. Transformer-based techniques seem to outperform the more traditional ML techniques showing better performance in common NLP tasks whilst being faster to train than LSTMs or RNNs. They solve the issue of vanishing gradients and show potential for few-shot learning.

Following these findings the following hypothesis is formulated:

- *Hypothesis 1: Transformer-based models will show better results than more traditional ML techniques because they show great potential in classifying student homework when dealing with only a few observations (few-shot learning).*

5.2 What are the properties of homework assignments from a data perspective?

There is no clear set of properties of homework assignments that can be used to indicate complexity to help choose the right ML model. To solve this problem the data set available from the case study is analyzed and the findings are translated into a set of properties. This is done using quantitative data analysis techniques and data exploration. The findings have been discussed with domain experts (teachers) and technical experts (in the field of NLP). In this case study, all homework assignments consist of open-ended questions that have to be answered in writing.

5.2.1 Structured v.s. Semi-structured

To answer the main question of this research there needs to be a better definition of the term ‘open questions’ first.

The dataset contains 15 open questions but these questions have different degrees of structure. For example, a student's answer could be one line of text that has to contain a specific keyword. On the other hand, a student's answer could contain his specific view on a subject which can lead to a higher diversity of answers within the dataset.

Question: Noem een voorbeeld van een beheerscyclus.
Answer: Plan-do-check-act cyclus

Example - structured question.

Question: Formuleer een scope voor dit project op basis van de informatie uit de casus.
Answer: Je gaat wel een inzending maken met daarin ideeën om bewustwording te creëren op over het onderwerp 'enkelvoudige fietsongevallen' onder de oudere fietser en/of weggebruikers. Hierbij gaat het over ideeën die de maatregelen en risico's in kaart brengen, waarbij begrip, bewustwording en herkenbaarheid ontstaat.

Example - semi-structured question.

The chosen dataset scores these open questions via a Rubric that supports teachers in deciding how many points are given. Most questions can score from 0 to 5 points but for a few questions, a student can score up to 20 points. For the questions that go up to 5 points, each point can be seen as a different category and the Rubric contains specific indicators that are required to score a certain amount of points.

5.2.2 Garbage in Garbage out

When looking at the distribution per question (figure 6) it shows that this dataset contains unbalanced data. For example, questions 3, 4, 11, 12, and 13 have all been either scored correctly (all points) or incorrectly (zero points). The other labels are rarely chosen. For the other questions, some labels are less representative. For these labels, it will be harder to predict the right class. In consultation with the teacher who made this exam (domain expert), the decision has been made to drop these less-represented labels (n < 10).

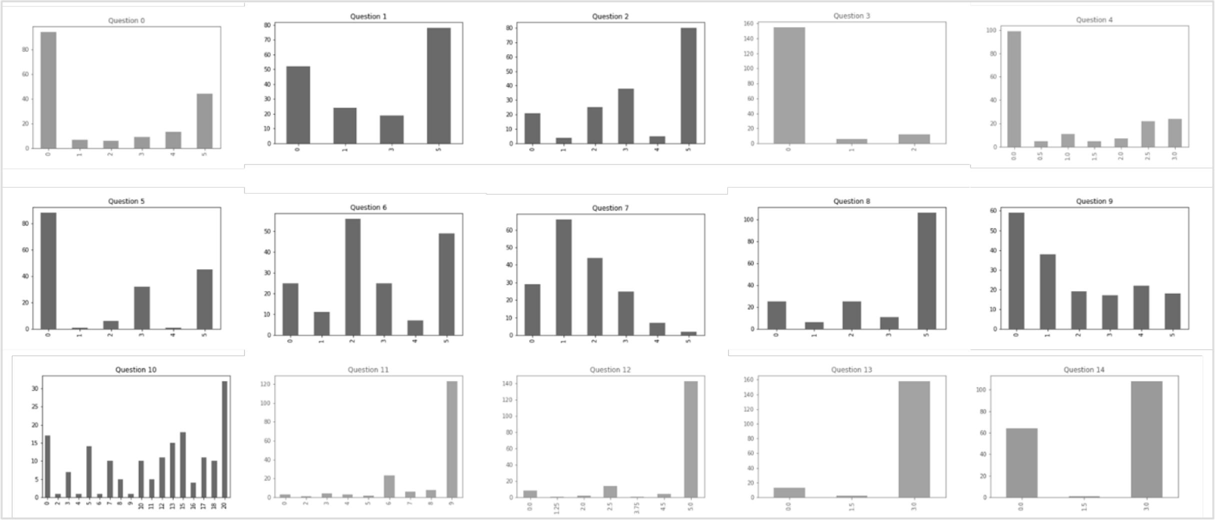


Figure 6: Distribution per question starting with questions 0 to 14.

Another obvious observation is the skewness of the data. For some questions (questions 4, 5, 11, 12, and 13) the data distribution is skewed, meaning almost all observations fall into the same class. It will be almost impossible to create a good model to automatically label these questions because even a naïve model would already show high performance and accuracy. For example question 3 a model could always use label 0 and already have an accuracy of above 95%. Because of this issue, these questions have been left out of the study.

Finally, the dataset contained one question (question 10) where students were asked to draw a diagram. Although drawing a diagram might be a common type of homework assignment it is outside the scope of a textual 'open question'.

After this cleaning step, the following questions have been selected to analyze and test with an ML model: 0, 2, 5, 6, 7, 9, 14. For a full description of these questions see Appendix A.

5.2.3 Question Complexity

The complexity of the question can indicate how hard it will be to decide what label an answer should get. To determine question complexity we look at the questions from a statistical angle and a theoretical angle by grouping the questions in the corresponding taxonomy level of bloom (Bloom, 1956).

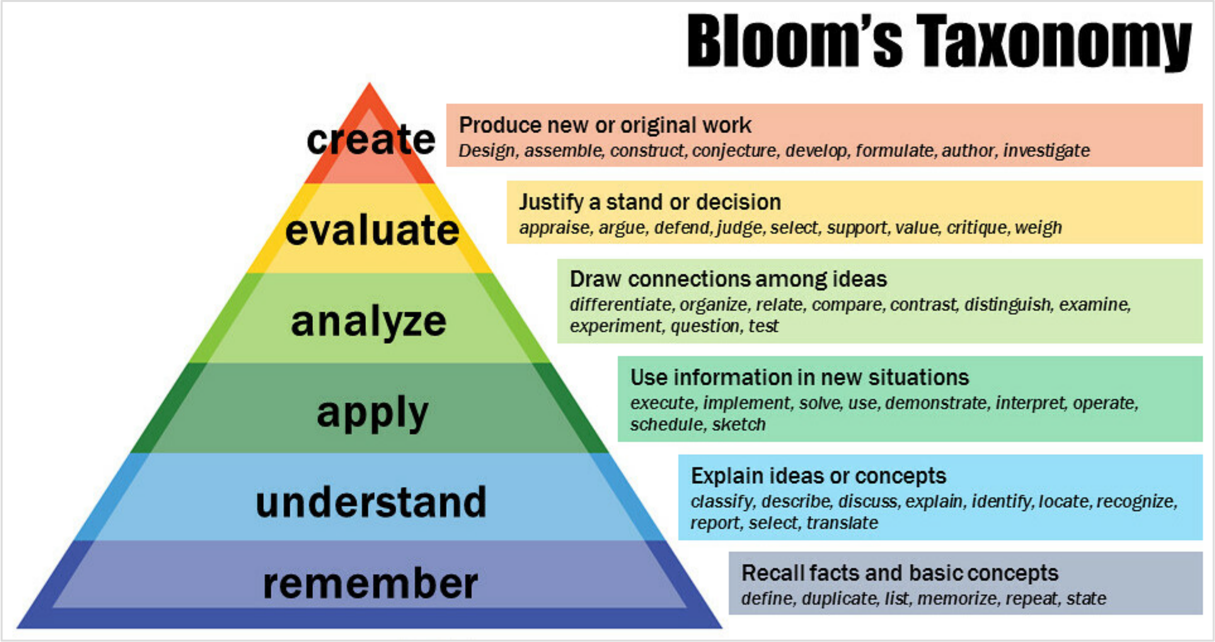
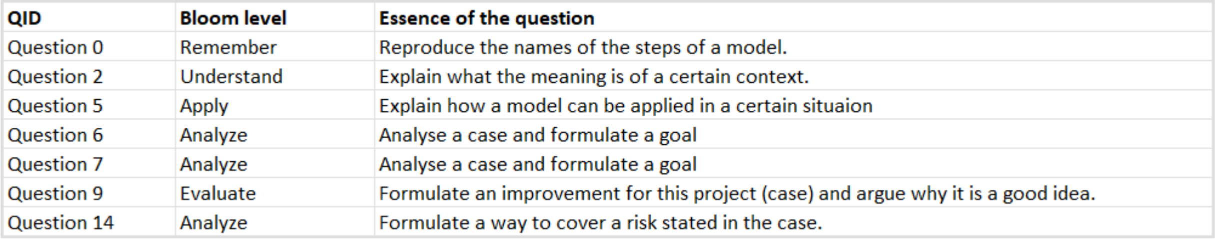

Bloom's Taxonomy
create
Produce new or original work
Design, assemble, construct, conjecture, develop, formulate, author, investigate
evaluate
Justify a stand or decision
appraise, argue, defend, judge, select, support, value, critique, weigh
analyze
Draw connections among ideas
differentiate, organize, relate, compare, contrast, distinguish, examine, experiment, question, test
apply
Use information in new situations
execute, implement, solve, use, demonstrate, interpret, operate, schedule, sketch
understand
Explain ideas or concepts
classify, describe, discuss, explain, identify, locate, recognize, report, select, translate
remember
Recall facts and basic concepts
define, duplicate, list, memorize, repeat, state

Image 5 - Bloom's Taxonomy visualized (Bloom, 1956)

Bloom's Taxonomy

The Taxonomy of bloom is a popular framework used for the classification of education learning objectives. This framework was chosen as a technique to classify each question with a corresponding bloom level because it is a popular technique within the organization that supplied the case study.

In the following table (table12) the questions are marked with their corresponding Bloom level (image 5). These levels might be good indicators of question complexity and in that sense be a good indicator of how hard it will be to automatically classify student homework assignments for each question.

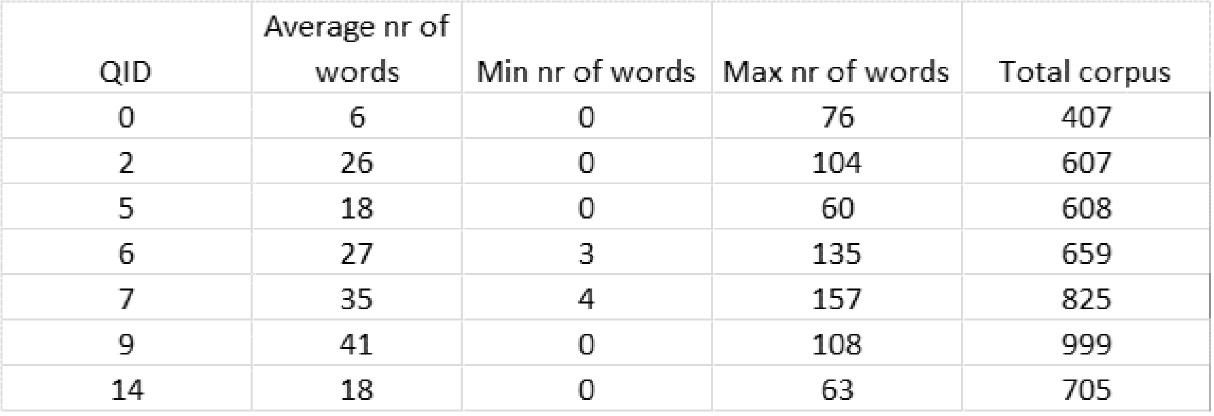


QID	Bloom level	Essence of the question
Question 0	Remember	Reproduce the names of the steps of a model.
Question 2	Understand	Explain what the meaning is of a certain context.
Question 5	Apply	Explain how a model can be applied in a certain situaion
Question 6	Analyze	Analyse a case and formulate a goal
Question 7	Analyze	Analyse a case and formulate a goal
Question 9	Evaluate	Formulate an improvement for this project (case) and argue why it is a good idea.
Question 14	Analyze	Formulate a way to cover a risk stated in the case.

Table 1: Questions with their corresponding bloom level.

Statistical properties

For the complexity of the question, the number of words used per answer is presented (table 2). In this table, you can see the average number of words used compared to the total words used (corpus) per question. These numbers of words indicate the complexity of the answers given by students and thus can indicate how hard it will be to predict the correct label.



QID	Average nr of words	Min nr of words	Max nr of words	Total corpus
0	6	0	76	407
2	26	0	104	607
5	18	0	60	608
6	27	3	135	659
7	35	4	157	825
9	41	0	108	999
14	18	0	63	705

Table 2 - Words per question

It looks like questions 8 and 9 contain the longest answers and the most complex corpus. Question 14 has a larger corpus but a shorter average answer length.

This length of an answer and/or the size of the corpus might be a good indicator of how well a model can classify the student's answers.

5.2.4 Keywords

Next to these statistical approaches, just looking at the data can give some insight into what types of techniques are required to successfully classify these assignments. Maybe there are specific keywords that a teacher looks for in an answer that can indicate how the answer will be scored. This showed that some questions (table 4) contain specific terms that a student needs to reproduce whereas other questions (table 5) have a great overlap in keywords between wrong and correct answers.

Most used words correct answers	
Word	***Times***
forming	40
norming	37
performing	35
storming	35
adjouring	21
adjourning	15
Samenwerking	6
project	6
stroming	5
fase	5

Most used words wrong answers	
Word	***Times***
fase	11
realistisch	8
project	8
meetbaar	8
Specifiek	7
elkaar	6
acceptable	6
kennen	5
leren	5
resultaat	5

Table 4 - Counting words for question 0

Question 0 has only 2 words that overlap. The words: ‘project ‘ and ’fase’.

Most used words correct answers	
Word	***Times***
fietshandelaren	54
meewerken	26
mee	25
werken	19
helmen	19
geven	18
fiets	17
willen	15
ouderen	14

Most used words wrong answers	
Word	***Times***
fietshandelaren	29
mee	11
fiets	11
ouderen	11
fietsen	10
helm	9
moeten	8
geven	7
risico	7

Table 5 - Counting words for question 14

Question 14 has many words that overlap between correct and incorrect answers. In total 5 words are overlapping and that is not even considering words like ‘helm’ and ‘helmen’ (which is a plural of ‘helm‘ in dutch. The word ‘fietshandelaren’ is the most common word in correct and incorrect answers.

5.2.5 Partial conclusion

It looks like some clear indicators for question complexity might give insight into how hard it will be for a model to automatically classify an answer.

The results from this research question are discussed with different domain experts: The teacher behind the exam and multiple people with knowledge about machine learning and the domain of NLP. From this short evaluation, it looks like the following properties are important properties that can be used as input for the development of a prototype.

Question quality

Some questions had to be dropped because the quality of the data was insufficient. For a question to be of high quality, there needs to be a certain data distribution where the data is not too skewed and all labels have a minimum of observations.

Question keywords

Questions that have a clear set of keywords that are common in a correct answer (and uncommon in a wrong answer) will probably be easier to automatically classify. Maybe statistical techniques will already show promising results without the need for more complex ML techniques.

Question complexity

The question complexity can be described using the answer length and the total corpus used. From this point on the complexity of a question is defined as:

- *Complexity = Average number of Words * Total Corpus*

Another indicator of complexity might be the level of the bloom taxonomy that is assigned to the question.

Following these findings, the following hypothesizes are formulated:

- *Hypothesis 2: If the question complexity increases it will be harder for an ML model to automatically classify student homework.*

- *Hypothesis 3: The higher the bloom level the harder it will be for an ML model to automatically classify student homework.*

5.3 What type of machine learning model is well suited to classify homework assignments?

To answer this question, multiple prototypes were developed for different questions to determine what types of questions are easy and what types of questions are hard to classify. First, the specific choices are explained, and then the results are evaluated.

Based on the previous findings 3 different types of models were chosen (explained further down this chapter) to develop multiple prototypes.

- Model 0: A baseline model: TF-IDF + SVM
- Model 1: RobBERTje (distilled)
- Model 2: RobBERT (larger)

5.3.1 Chosen techniques

Based on the results as described in chapter 5.1 the following techniques have been chosen to develop the prototypes. First, the techniques for the baseline model are explained and then the techniques for the modern ML model are explained.

Model 0: A Baseline Model

This model serves as a baseline to compare against the other 2 models. It will put the performance of the model in a context where it is easier to explain the significance of values. There were many options available to build this baseline model but the following techniques have been chosen.

TF-IDF

TF-IDF, short for Term Frequency-Inverse Document Frequency, is a statistical method to look at words in a document. It assigns a value to each word based on how often a word appears in a document compared to how common that word is in a set of documents (corpus). TF-IDF was used to tokenize (vectorize) with an n-range of 1 and 2. Before tokenizing the text it was cleaned using Lemmatization, removing Dutch stopwords, and transforming the words to lowercase. No action was taken to correct spelling or handle other anomalies such as slang or abbreviations (Kowsari et al., 2019).

TF-IDF was chosen based on the results from chapter 5.1 because it is a common embedding technique in the category of traditional word embeddings.

SVM

In a small prototype, SVM was compared to Random Forest and Naïve Bayes and showed the most promising results. To find a good baseline a grid search was used to find the correct configuration for the baseline model.

SVM was chosen because it was one of the most popular text classifications found in chapter 5.1 and performed better than an RF or NB.

LSTM was a good candidate to use as a baseline model but LSTMs require a lot of data to be trained to get good results which are not available with the current case study and for that reason will not be well suited to work as a baseline model.

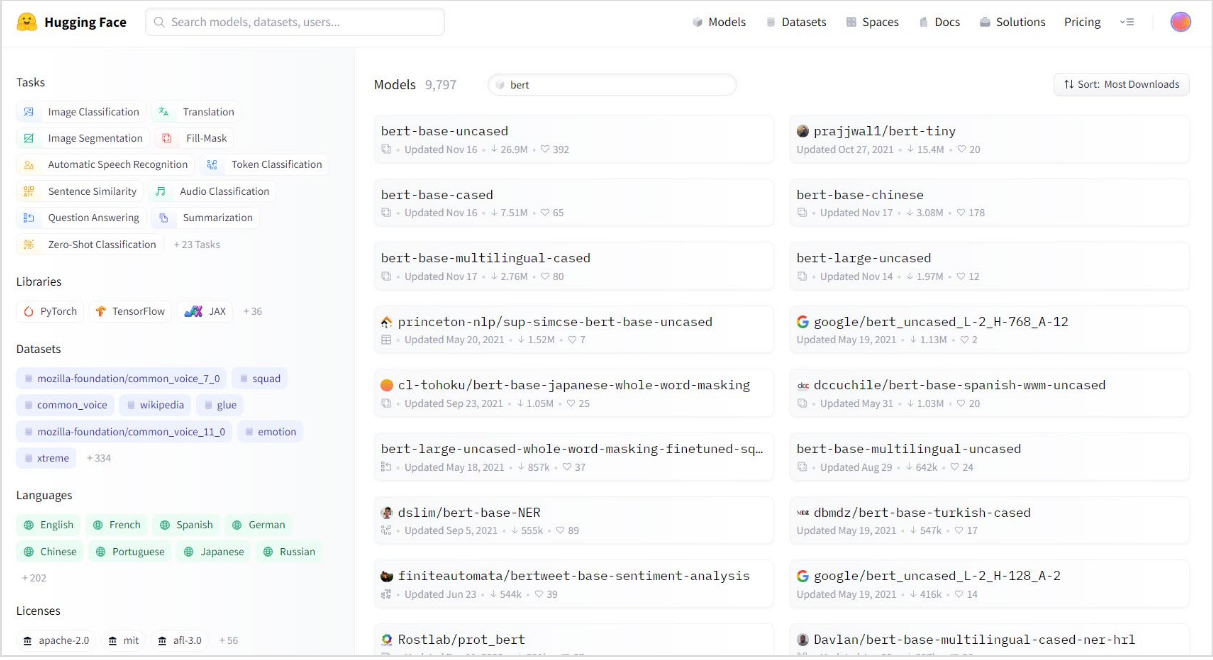
Model 1: RobBERTje

A BERT-like model was chosen because it occurred often in previous research (chapter 5.1) and models are available that are pre-trained on a dutch corpus. Compared to alternatives such as GPT BERt-like models show more potential when dealing with classification whereas GPT shows more potential when dealing with next-sentence predictions.

BERT: The original architecture

The chosen pre-trained model is derived from the original BERT architecture (Devlin et al., 2018b). Based on this architecture many models have been developed and pre-trained on a specific corpus. These models can easily be accessed via the HuggingFace Library (currently containing over 90,000 pre-trained models) (image 6).

The BERT architecture was chosen because of its specific Dutch models (for example BERTJE) and its performance on classification tasks (compared to alternatives such as GPT, chapter 5.1.3). Although BERT has a maximum input of words, this will not be an issue with the current dataset as the longest answer has a length of 157 words (chapter 5.2.3).



bert-base-uncased
Updated Nov 16 • ↓ 26.9M • ♡ 392

prajjwal1/bert-tiny
Updated Oct 27, 2021 • ↓ 15.4M • ♡ 20

bert-base-cased
Updated Nov 16 • ↓ 7.51M • ♡ 65

bert-base-chinese
Updated Nov 17 • ↓ 3.08M • ♡ 178

bert-base-multilingual-cased
Updated Nov 17 • ↓ 2.76M • ♡ 80

bert-large-uncased
Updated Nov 14 • ↓ 1.97M • ♡ 12

princeton-nlp/sup-simcse-bert-base-uncased
Updated May 20, 2021 • ↓ 1.52M • ♡ 7

google/bert_uncased_L-2_H-768_A-12
Updated May 19, 2021 • ↓ 1.13M • ♡ 2

cl-tohoku/bert-base-japanese-whole-word-masking
Updated Sep 23, 2021 • ↓ 1.05M • ♡ 25

dccuchile/bert-base-spanish-wwm-uncased
Updated May 31 • ↓ 1.03M • ♡ 20

bert-large-uncased-whole-word-masking-finetuned-sq...
Updated May 18, 2021 • ↓ 857k • ♡ 37

bert-base-multilingual-uncased
Updated Aug 29 • ↓ 642k • ♡ 24

dslim/bert-base-NER
Updated Sep 5, 2021 • ↓ 555k • ♡ 89

dbmdz/bert-base-turkish-cased
Updated May 19, 2021 • ↓ 547k • ♡ 17

finiteautomata/bertweet-base-sentiment-analysis
Updated Jun 23 • ↓ 544k • ♡ 39

google/bert_uncased_L-2_H-128_A-2
Updated May 19, 2021 • ↓ 416k • ♡ 14

Rostlab/prot_bert

Davlan/bert-base-multilingual-cased-ner-hrl

Image 6 – Example of a few BERT-like models on the Hugging Face repository.

RobBERT and RobBERTje

Specifically, RobBERTje (Delobelle et al., 2021) has been chosen as the pre-trained model for the classification task. It can be used on a large number of Dutch natural language processing tasks and because of its small size (distilled), it’s faster while still achieving close to state-of-the-art results (Delobelle et al., 2022). This model is based on the RoBERTa architecture (Liu et al., 2019) and specifically, a Dutch version of RoBERTa called RoBERT (Delobelle et al., 2020). It’s a distilled version of the original model to decrease the size and increase training time while keeping high performance.

The original RoBERT is trained on 39GB of Dutch corpus containing 6.6B tokens (words). The model shows great potential when working with small datasets (Delobelle et al., 2020). The distilled version of the model (RobBERTje) is a small version of the original model that is trained to mimic the larger model. In comparison, RobBERTtje has 74M parameters compared to the original 117M parameters of the RoBERT model.

Because of its smaller size, it might have a lower performance but is great for prototyping. After creating a prototype for every question with this model the larger model (RobBERT) is tested to compare performance.

Transformer library

To build the model the transformer library was used to load and build the models. The library also supplied the default classification layer that is recommended to be used when working with BERT. The output of BERT are vectors with the size 768 containing information (features) about the sentence. This information serves as the input to predict the right label for a question. The classification layers turn these vectors into one vector containing a prediction for every label. So this library that creates the classification layers needs to know how many labels it’s going to predict.

Train/test split

A default split of 60/20/20 was used to create a train, validation, and test set. Previous research shows the importance of choosing the right validation method (Vabalas et al., 2019) and other strategies such as K-FOLD Cross-Validation might show better results. But for the scope and goal of this research for all models the same train/test split was used. The dataset was split into the following 3 groups:

- Training: 60%
- Validation: 20%
- Testing: 20%

During the development of the prototypes, the training and validation set were used. In chapter 5.3.5 you can see the results of the final test set (data that has not been seen by any model thus far).

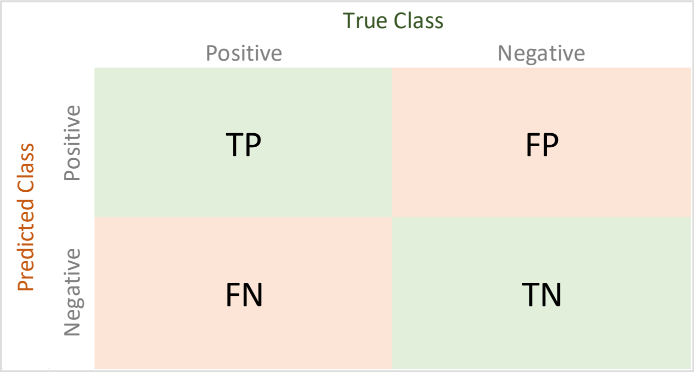
Loss function

No loss specified in compile() - the model's internal loss computation provided by the HuggingFace library will be used as the loss. This is a common way to train TensorFlow models in Transformers. By default, this is the cross-entropy loss function.

Accuracy and F1 Score

The performance of the model was determined by calculating the accuracy and F1 score. These metrics were chosen based on the chosen metrics in previous research. Accuracy is a metric that measures the percentage of correct predictions made by a model. It is calculated by dividing the number of correct predictions by the total number of predictions. But because some questions have a limited sample size and imbalanced labels, the accuracy will not always be a good indicator. F1 score is a more robust metric than accuracy, especially when dealing with imbalanced datasets, as it takes both precision and recall into account. Accuracy, Recall and, F1 score is calculated using the predicted class and true class (Figure 7).

- *Accuracy = TP / (TP + FP)*
- *Recall = TP / (TP + FN)*
- *F1 = (accuracy * recall) / (accuracy + recall)*



		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Figure 7: Confusion Matrix with predicted class vs true class and formula's to calculate accuracy and F1 score.

Model configurations

The following configurations are used to build the prototype. The models were tweaked a few times on the train and dev set before finally running it on the not previously seen test set.

Model 0: A baseline model: TF-IDF + SVM

The baseline model is built up in 3 steps: cleaning the data, embedding with TF-IDF, and classification with SVM. The model is trained and the train and validation set and finally the performance is calculated via the test set.

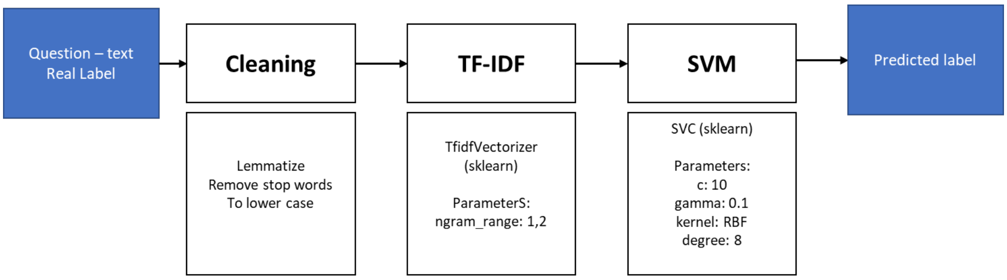


Figure 8: Model 0 configuration

Model 1: RobBERTje (distilled)

The BERT model is built using the HuggingFace library. The following pre-trained model is used to build the sequence classifier.

- DTAI-KULeuven/robbertje-1-gb-non-shuffled

The model is trained on the train and validation set for 10 epochs and finally, the performance is calculated via the test set EarlyStopping (Keras library) is used to prevent overfitting and to select the best model during training. On average the best model (lowest validation loss) was found around epoch 6 but training was continued for a few more epochs to be sure the best model was found.

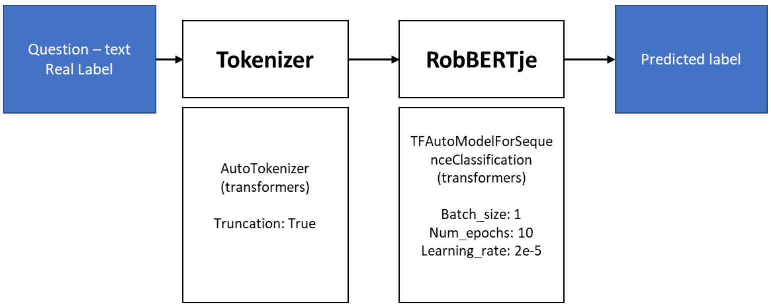


Figure 9: Model 1 configuration

Model 2: RobBERT (larger)

Model 2 uses the same configuration as model 1 except for the following pre-trained model that is used:

- pdelobelle/robbert-v2-dutch-base

5.3.2 Baseline model versus RobBERTje

The baseline model (model 0) is tested against the RoBERTje model (model 1). The performance of the model is calculated using its performance and F1 score on the test set. For every question, a baseline model and RoBERTje model are trained (with the same configuration) and for every model, the performance (figure 10) and F1 score (figure 11) are charted.

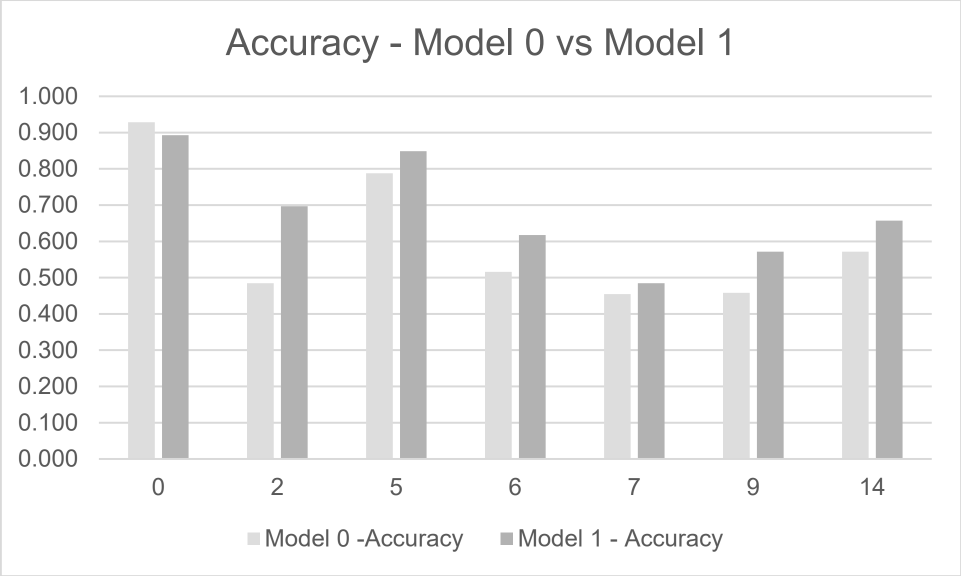


Figure 10: Accuracy Model 1 vs Baseline Model 0

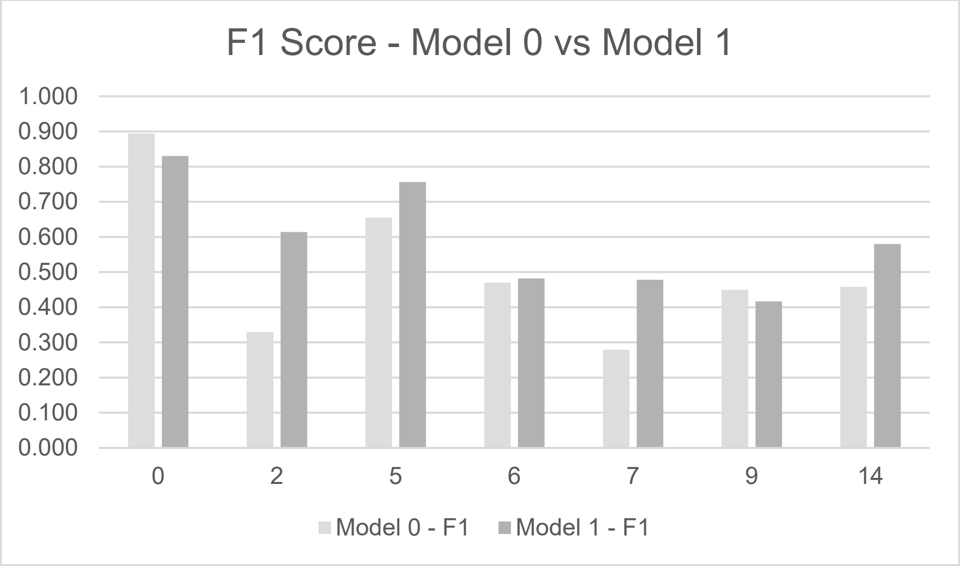


Figure 11: F1 Score Model 1 vs Baseline Model 0

Some models show great potential straight off the bat. For questions 0 and 5 both model 0 and model 1 show a good average and F1 score. This is without any model optimization for that specific question.

All the questions except question 1 show an increase in accuracy for the BERT-like model compared to the baseline model. This is the same for the F1 score except for question number 9. This could be related to the number of labels and the distribution of the labels which is better reflected in the F1 score (table 6). Questions 6, 7, and 9 show accuracy below 50% making these questions the hardest to predict. Question 14 only barely shows an accuracy above 50%.

QID	Nr Of Labels	Model 1 - F1
0	2	0.830
2	4	0.614
5	3	0.756
6	5	0.482
7	4	0.478
9	6	0.417
14	2	0.580

Table 6: Number of labels compared to F1 score for model 1 with question 9 marked with the lowest F1 score.

5.3.3 Larger models

For model number 2 the larger model named RobBERT was used to see if the size of the model would significantly increase the performance. For both model 1 (smaller) and model 2 (larger), the performance (figure 12) and f1 score (figure 13) are charted.

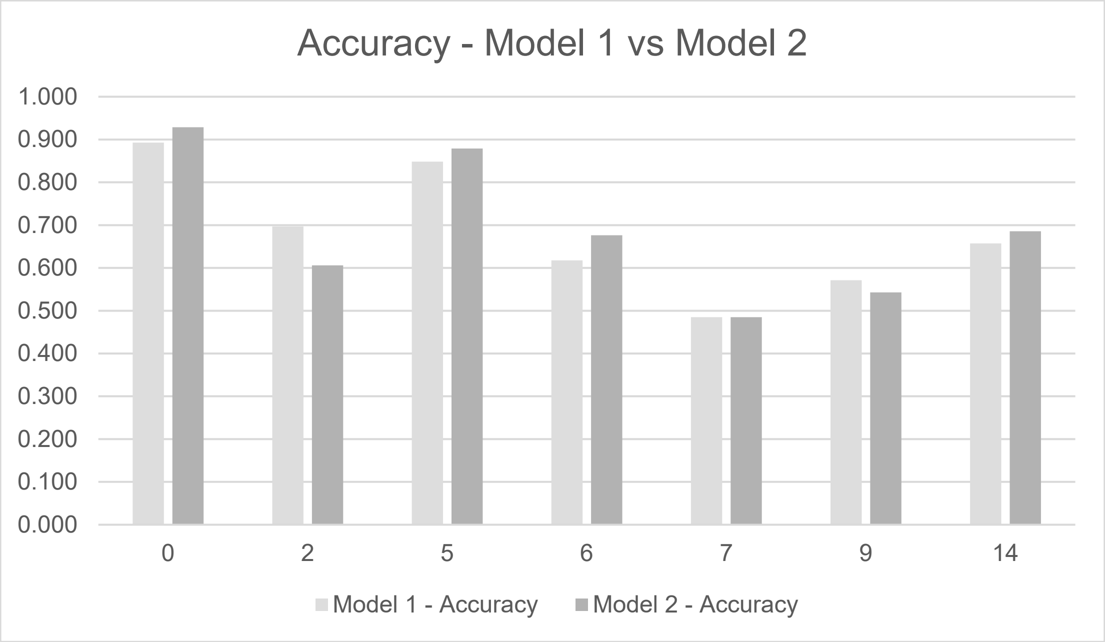


Figure 12 - Accuracy Model 1 vs Model 2

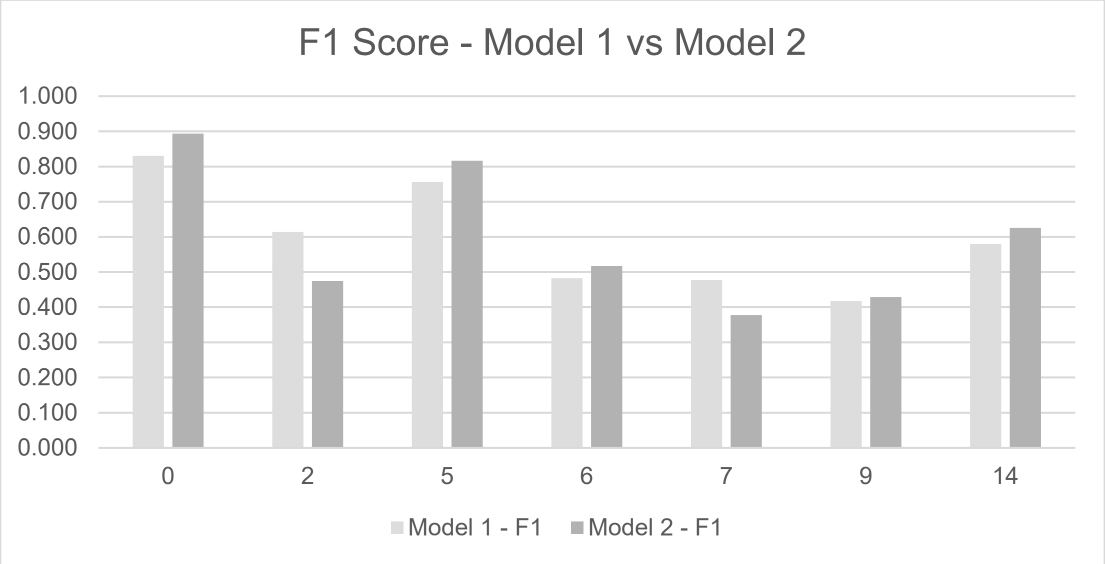


Figure 13 - F1 Score Model 1 vs Model 2

Almost all questions show a small increase in accuracy score and F1 score except in a few cases. The F1 score dropped for question 7 and the accuracy dropped for question 9.

Question 2 seems to decrease in both F1 score and Accuracy.

5.3.4 Prototype evaluation

Next to calculate the performance of the model the prototypes were evaluated by organizing a focus group. This focus group consisted of 7 technical experts in the field of machine learning and specifically in the field of NLP.

The goal of the focus group was to determine if the chosen techniques (chapter 5.3.1) were correctly applied to develop the prototype. The focus group was an online event where first the goal of the research was explained. Then the chosen techniques were explained and the results (chapter 5.3.1) were presented. After this short introduction, the following questions were discussed with the group.

The following questions were formulated and discussed during the focus group with a short summary of topics that arose during the focus group.

1. Do you see any problems or opportunities with the current approach and chosen techniques in the context of classifying student homework assignments?

   The evaluation showed that the technical use of BERT was correct, but to improve the accuracy of the more complex questions (questions 7 and above), some adaptations to how the BERT model was used are needed. These possible adaptations are found in chapter 6.3 (Discussion). Currently, BERT is used in the context of text classification but in regards to checking student homework assignments a BERT can be used to compare two sentences (student answer and a predetermined correct answer).

2. Are there any options to improve the performance of the model regarding data preparation (such as data augmentation) in the context of working with a small unbalanced dataset?

It might be possible to improve the performance of certain questions by duplicating answers with only a few labels. Up and downsampling might help with dealing with an unbalanced dataset.

3. The current solution uses a classical train/test split. Are there other types of train/test splits worth testing that could improve the performance?

Cross-validation is a popular technique when working with a small dataset. But it is not very common to use this technique when working with models such as BERT.

4. Do you see any low-hanging fruits in the current solution that might improve the performance of the model?

Currently, the model is relatively small compared to other models (chapter 5.1.4). Larger models might show better results when dealing with few-shot learning. The current BERT model that is used is trained on dutch data. Multilingual models might perform even better.

5.3.5 Partial conclusion

The performance of these models is compared to the complexity of the answers (Figure 14). As stated previously the complexity is calculated by multiplying the average answer length by the number of words used in the corpus (chapter 5.2.5).

This shows that there might be a correlation between the number of words used in an answer, the total corpus, and the accuracy of the model.

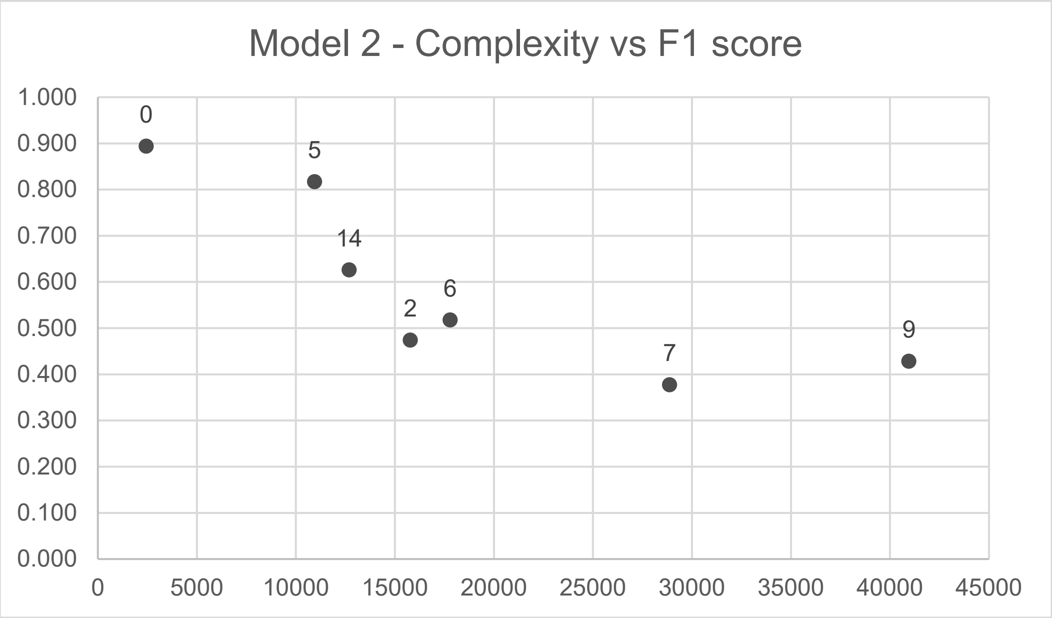


Figure 14: Model performance versus question complexity with question id as the label.

With these results, the hypothesis from chapters 5.1 and 5.2 can be reviewed.

Hypothesis 1: Transformer-based models will show better results than more traditional ML techniques because they show great potential in classifying student homework when dealing with only a few observations (few-shot learning).

The BERT-based model shows an increase in accuracy for almost all questions (except question 0).

Hypothesis 2: If the question complexity increases it will be harder for an ML model to automatically classify student homework.

As stated previously there might be a correlation between question complexity and the performance of the model (figure 14).

Hypothesis 3: The higher the bloom level the harder it will be for an ML model to automatically classify student homework.

Questions 6, 7, 9, and 14 are questions with a higher bloom level (chapter 5.2.3, table 1). Question 14 shows relatively good performance (compared to the other questions) but question 9 (with the highest bloom level) shows poor performance.

6. Conclusion, Discussion, and Recommendations

Following the results of this research a conclusion can be drawn. The goal of this research was to develop a set of guidelines for future developers that want to automatically classify student homework assignments with their datasets. You can find these guidelines in chapter 6.2. But because of the constraints of this research (such as time and knowledge) a large chapter with points of discussion is added. These points of discussion have to be taken into consideration when reading the conclusion and the guidelines. The topics in this chapter point out the limitations of this study. Finally, a set of recommendations is formulated in the form of future work. This consists of topics that might be interesting for future research.

6.1 Conclusion

Modern ML techniques show potential in automatically classifying student homework assignments. But with only a small dataset classic BERT-like classification models do not always show great performance increases compared to more traditional ML techniques such as TF-IDF and SVM.

The data analyses from the case study show that questions have different types of complexity. There seems to be a correlation between this complexity and the performance of the classification task (figure 16).

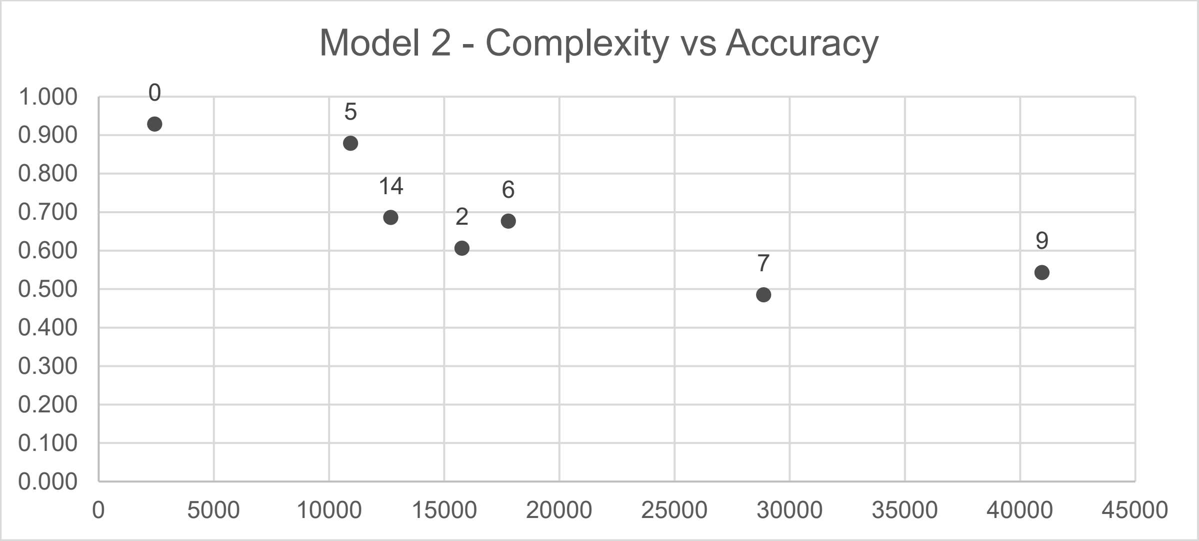


Figure 16: Complexity versus Accuracy.

In total 5 questions scored an accuracy of over 60% (higher than chance) without hyperparameter tuning or any other type of ways to increase the performance. These questions show potential to be used in a formative feedback test and can be classified automatically. But before this can be applied in a real-life setting other performance-increasing possibilities have to be tested.

Next to the raw performance, the data distribution needs to be taken into account. Some questions only have a few labels to predict but for questions with more labels, the accuracy can show a distorted picture. For these questions, it is better to look at the F1 score (Chapter 5.3.5, Figure 14).

The complexity of the question regarding the level of Bloom does not seem to be a good indicator in this case Study. Questions 7 and 9 are at a higher level of Bloom (Analyze and Evaluate) and it seems these questions are hard to predict. But question 2 is in one of the lower levels of Bloom (Understanding) but still shows poor performance in this case study.

Taking all of this into account the main research question can be answered:

How can semi-structured open-ended *questions in student homework assignments automatically be classified by a model that is trained on a small dataset (n=~100) using modern ML techniques as to provide students with formative feedback without costing more time?*

It is possible to automatically classify student homework assignments with modern ML techniques. In this case study a BERT-like model is used for this classification task showing promising results for certain types of questions. Unfortunately, not all types of questions show the same high accuracy. In conclusion, the proposed solution will not work in a real-life setting for all types of student homework assignments.

6.2 Guidelines for future developers

If you want to build your own solution to automatically classify student homework assignments, the following guidelines may assist you in that process. These guidelines are grouped by the categories in the design science methodology.

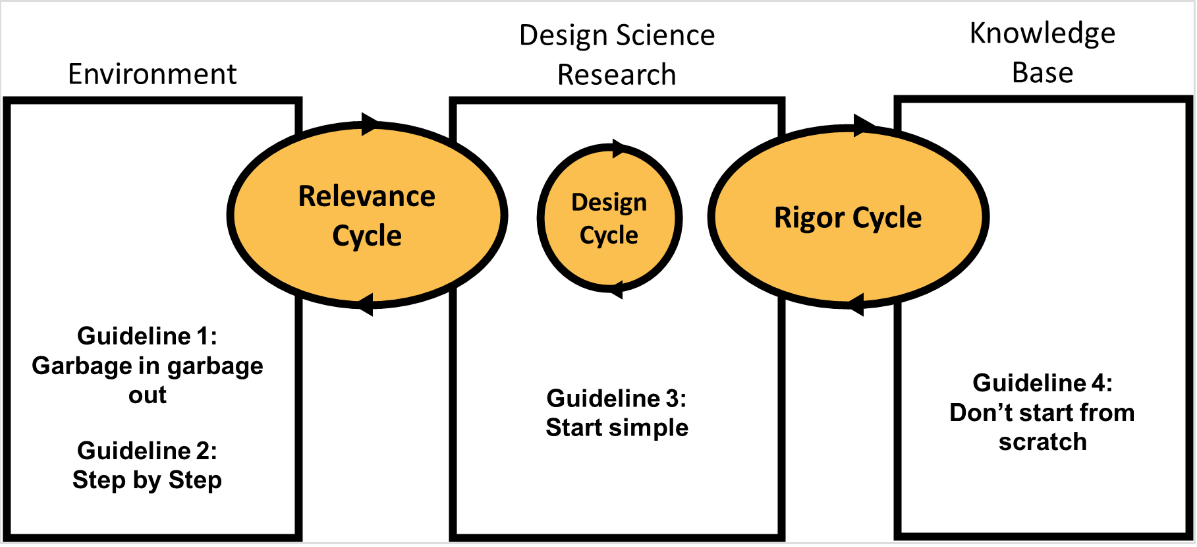

Environment
Design Science Research
Knowledge Base
Relevance Cycle
Design Cycle
Rigor Cycle
Guideline 1: Garbage in garbage out
Guideline 2: Step by Step
Guideline 3: Start simple
Guideline 4: Don’t start from scratch

Figure 17: Guidelines grouped by the design science methodology.

Guideline 1: Garbage in garbage out

Check if the homework assignment or test you want to use as input is of a high enough quality. Questions that show skewed results (always correct or incorrect) are almost impossible to work with. There are a few standards out there to check the quality of a test such as calculating a p-value or Cronbach's alpha. Select only the questions or assignments that are of high enough quality before starting development on a prototype.

This guideline was formulated based on the results of the data analysis (chapter 5.2). The results showed that in some questions in the case study, the results were skewed. This means certain labels are underrepresented.

Guideline 2: Step by Step

Do not try to implement a solution without rigorous testing. Even if a pilot project is successful, human oversight (as recommended by the European guidelines on ethics in artificial intelligence) (European Commission & Directorate-General for Communications Networks, 2019) should always play a role in providing students with feedback, and a student should not in any way be disadvantaged by an automatic feedback system.

This guideline was formulated based on the results of the literature review (chapter 5.1), the prototypes (chapter 5.3), and the interviews with domain experts (chapter 6). The performance of the model is far from perfect and this research does not include any ethical considerations that are relevant when applying this model in a real-life situation.

Guideline 3: Start simple

Traditional ML models show promising results on questions with low complexity. First, try more traditional solutions such as a bag of words (BOW) or more sophisticated techniques such as TF-IDF. These techniques are faster to train and easier to explain how they work (explainable AI). If the previous solutions do not show promising results, more complex solutions based on deep learning like BERT might solve the issue.

This guideline was formulated based on the results of the prototypes (chapter 5.3). Even though the model based on the BERT architecture showed an increase in performance for most questions the original baseline model with SVM and TF-IDF already showed great accuracy.

Guideline 4: Don’t start from scratch

If you choose to go down the road of deep learning start with the right pre-trained model. There are many pre-trained models to choose from (take a look at HuggingFace) with even specific models that are trained on the corpus of a specific domain.

This guideline was formulated based on the results of the literature review (chapter 5.1) and the prototypes (5.3). There were many models to choose from to develop this prototype. An important side note is that in this research these pre-trained models were not compared against custom build deep learning solutions such as LSTM.

6.3 Discussion

This research yielded topics that could not be further explored because of the scope. Some of these topics came forward after reviewing the results of chapter 5 and others after a set of interviews with experts in the domain of machine learning and experts in the domain of education. These topics have been explored briefly and described in this chapter. It can give a better insight into some of the limitations of this study.

The topics are grouped by the original design science methodology and finally, threats to the validity of this research are presented (figure 15).

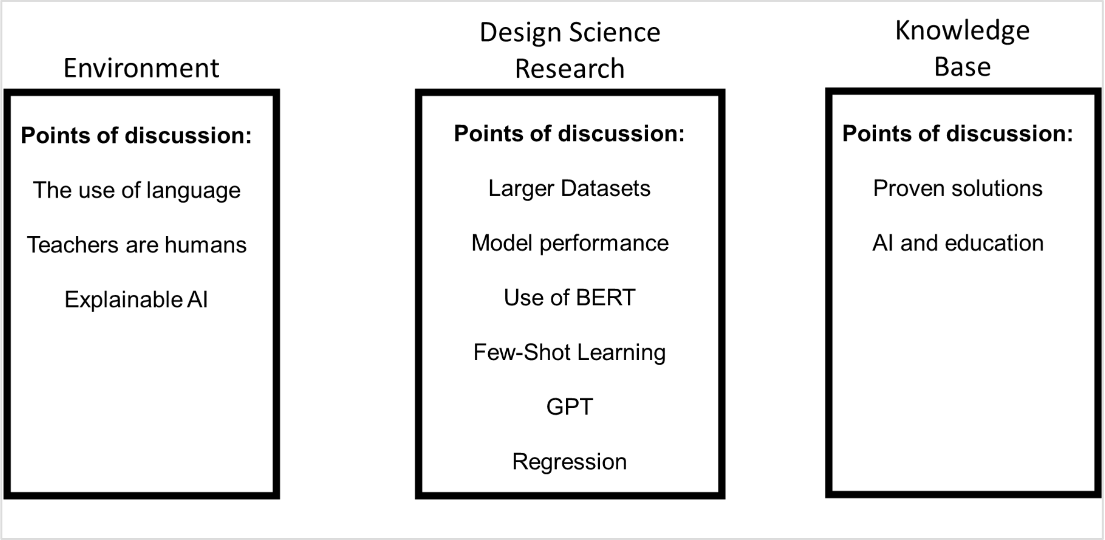


Figure 15. Points of discussion grouped by methodology.

6.3.1 Environment

This research focuses on the possibility to apply ML to automatically classify student homework but does not consider all the problems that may arise with a real-world application. The following topics explore potential issues that might arise when applying these types of models in a real-world environment.

The use of language

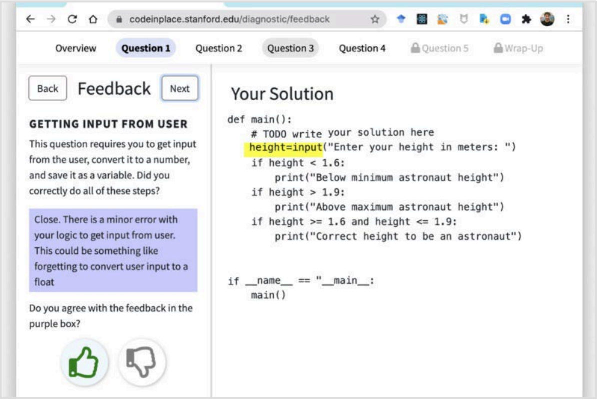
The dataset currently only contains the given answer as text and the given label by a teacher. It does not contain any background information on the student or information about the quality of the text. If, for example, a text contains more spelling errors, a teacher might label the answer as incorrect. In a conversation with the original teacher who coördinated the exam (case study), it was indicated that spelling errors in answers were ignored if the answer was still technically correct but it was uncertain if all teachers checked the answers in the same way. Next to grammatical errors, an answer can be correct but written in a cumbersome way. If a model learns that cumbersome language is often labeled as incorrect, it might pick up on unintended signals that should not determine the right classification of an answer.

Teachers are just humans

The dataset was labeled by multiple teachers, but it was not taken into account what teacher labeled what question when working with the given dataset. There is a possibility that there is a large inconsistency between teachers in how they grade a document and sometimes a teacher makes a mistake (yes, he is just a human after all!). This inconsistency was not taken into account for this research but may influence the results and the performance of the created prototypes.

Explainable AI?

Although this prototype (and potentially future implementations) do not process personal data, the automatic decision still has an impact on a person (the student taking the test). Compared to more statistical models, deep learning models such as BERT are a black-box solution. It’s hard to determine why a specific prediction was made on a specific example.
This needs to be taken into consideration when developing a solution based on these prototypes. When implementing its prototype for the automatic classification of student homework, Stanford University only used its automatic grading system for reviewing homework and not the final exam (Wu et al., 2021). Next to that, they gave the students a way to indicate if they agreed with the given feedback on their work (image 7).



GETTING INPUT FROM USER

This question requires you to get input from the user, convert it to a number, and save it as a variable. Did you correctly do all of these steps?

Close. There is a minor error with your logic to get input from user. This could be something like forgetting to convert user input to a float

Do you agree with the feedback in the purple box?

```
def main():
    # TODO write your solution here
    height=input("Enter your height in meters: ")
    if height < 1.6:
        print("Below minimum astronaut height")
    if height > 1.9:
        print("Above maximum astronaut height")
    if height >= 1.6 and height <= 1.9:
        print("Correct height to be an astronaut")


if __name__ == "__main__:
    main()
```

Image 7 - Student interaction with automatic grading system (Wu et al., 2021)

The EU guidelines on ethics in artificial intelligence (European Commission & Directorate-General for Communications Networks, 2019) state that there should always be human oversight when working with an AI model. A teacher should be able to override a decision made by the system or model. Insight into why a model made a certain prediction can support a teacher in when and how to override a given grade.
There are techniques available to get more insight into how a neural network makes a decision such as sensitivity analysis and layer-wise relevance propagation (Samek et al., 2017).

6.3.2 Design Science Research

During the development of the prototype and evaluation of these prototypes, some topics arose that have not been tried during this research. The following topics might be good additions to improve the quality of the prototypes.

Larger datasets

The current results show that some questions are harder to classify than others (such as question 9). A closer look at this question shows that this is a type of question where checking if an answer is correct requires way more information than just the presence of certain keywords (or a combination of keywords).

Question: Je mag voor dit project één idee inzenden. Welke fasering is het meest passend voor een dergelijk project en waarom?
Translation: You can submit 1 idea for this project. What phase would be best suited for your idea and why?

Example - Question 9

With only so few observations it is easy to suggest that the data just does not contain enough examples for the model to get a grip on what is important. More examples could help increase the performance of questions such as these, but that has to be determined in future work.

Increasing model performance

For the prototypes, the step of optimization was skipped. It is not the goal of this research to find the best-performing model but to see if ML can be used to classify student homework assignments. The performance of all these models could potentially be increased by tuning the hyperparameters of the model. There might be other ways to increase the performance but this is not included in the scope of this project.

Few- or Zero-Shot Learning

One common downside of working with neural networks is the large quantity of data that is required to train these models. Although the prototypes have been developed with a model that shows potential when working with small datasets, other strategies might show better results when working with a small dataset.

Few-shot learning is a technique that can help solve this problem. With few-shot learning, the model is optimized for working with only a small dataset. With zero-shot learning, the model is trained in a way that it can predict a class for data it has never seen before. As stated previously (chapter 5.1.4), large Transformer models should show promising results when working with a small dataset (FSL). Although these techniques sound promising, not a lot of research is available on applying these techniques to BERT-like models.

Use of BERT

There are many ways to use a BERT-based model. HuggingFace, fortunately, has good documentation on how to use these models. But working with transformers is still a new field with not a lot of good examples of how to implement the 'best' solution. The prototype that has been built may be a sub-optimal solution and will require more work before it can be turned into a viable solution that can be applied in a real-life setting. Although the prototype is well suited in this context, future work may show different outcomes based on how the prototype was developed.

Specifically, when working with text classification there might be solutions available that have not been applied to the current prototype. For example, concatenating the correct answer (answer of a teacher) to a given answer gives the model a better grip on what a correct answer looks like (it makes the words in the correct answer more important).

Next to using BERT for the classification, other NLP tasks might be better suited to solve the feedback problem. Next sentence prediction, sentence similarity, or masked language modeling are techniques that might be good alternatives to solve the problem of automatic feedback. For example, checking if an answer is similar to already previously checked correct answers. Although this sort of happens in the background by creating the vectors that contain information about the answer and then using these vectors to predict the correct label it might be possible that currently not the full potential of BERT is being used.

What about GPT?

Next to the BERT model used in the prototypes, there are a few other types of models that can be tested such as GPT-2 (Radford et al., 2019) or ELMO (Peters et al., 2018). It is hard to predict if these models can increase the performance of the classification task. Especially GPT-2 (or the newer GPT-3) (Brown et al., 2020) shows great potential for universal NLP tasks. Espiaccialy in the context of Few-Shot learning. The literature review showed that for a classification task, BERT was a good fit. But other models such as GPT show great promise when dealing with other NLP tasks. Future work might determine that these alternative models show more potential than BERT-based models.

Regression versus classification

The student work in the case study is classified by using a Rubric, but within this Rubric, the classes each have a score that determines how 'good' the student's answer is. Currently, the classification problem does not take into account this order of the classes. This could mean potentially important information is lost by treating this problem as a classification problem.
This issue can be solved by treating the problem as a regression problem and trying to predict the score that an answer should receive. This score can later be mapped to the appropriate Rubric class.

6.3.3 Knowledge Base

Based on the available knowledge in the literature some questions remain unanswered. The following topics might be a good addition to the current knowledge base.

Proven solution

The field of transformer-based models is relatively new and not every domain has copy-paste solutions available for common problems. The results shown in this research are only an exploration of the possibilities of applying these types of models in the domain of education. The results may aspire future developers to try certain solutions on their datasets. The chosen techniques in this research may not be the perfect fit for every problem within this domain. More research is needed to establish what solutions work best in certain situations.

The role of machine learning in education

This research focuses on the possibilities that modern ML techniques offer in automatically providing students with feedback on their work but it does not answer all the questions surrounding AI and education. Should AI help to improve the quality of feedback or should it decrease the time needed to provide students feedback? These types of questions are important to answer before ML techniques can be fully applied within an educational setting.

6.3.4 Research Validity

In this chapter threats to the validity of the research are presented and classified into the following categories: Construction, Internal and External (Wohlin et al., 2000).

Construction Validity

Because of the scope of this research (in the shape of time and knowledge) models were selected to develop a prototype that might not be best suited to answer the question. There are so many possible solutions to compare that there is a reasonable possibility that other techniques will show different (better) results with the same dataset and context.

Some results show that certain types of questions are harder to classify. But there are still many modern ML techniques that have not been tested. Specifically for the questions that are harder to predict (questions 7 and 9) other types of models or techniques might show better results.

Internal Validity

There seems to be a relationship between question complexity (defined as the total used corpus times average answer length) and the performance of the model (accuracy and F1). But corpus and average answer length are just statistical properties that might be affected by a third factor that has not been identified. In this research, one of these possible properties is inspected in the form of Bloom's taxonomy. But without a clear-cut definition of what defines question complexity it is possible that some critical properties have not yet been identified. This threat is formulated as a recommendation for future work (chapter 6.4).

External Validity

The results of this research have been reproduced in different settings but only on 1 dataset. It is hard to tell if other datasets with the same properties will show similar results. It could be possible that the results can be replicated by calculating the average number of words and the total corpus, and with the same model configuration, it is possible to check if the results of this research are valid.

For this reason, it is currently not possible to draw any hard conclusions outside the boundaries of these specific questions and this specific dataset.

6.4 Future works

This research is only an exploration of the use of machine learning in the context of education. More research is needed before a solution (as proposed in this research) is ready for a production environment.

Definition of an open question

Currently, there is no clear-cut definition as to what defines an open question and what the difference is between a structured open question versus an open format question. More research on this subject could help identify the complexity of a question. And in turn, this complexity can be a better indicator to determine how hard it will be for an ML model to automatically classify student homework.

For now, it is recommended to use statistical analytics such as counting words (BOW) and corpus length as an indicator to determine the level of structure of a question.

Larger models

This research does not include very large models such as GPT-3. These models show promising results in the domain of Few-Shot learning (chapter 5.1) and might be the key to solving some of the performance issues with the classification of complex questions.

Unfortionaltly these larger models are not publicly available. And their size makes them harder to use which makes prototyping more difficult.

Continue with the exploration

The field of NLP and techniques such as transformers and BERT are fairly new and more research is needed on this topic (specifically in the field of education) before reliable solutions are available. This exploration contains a lot of topics that require more attention ranging from technical topics such as data augmentation to ethical questions about the quality of feedback. Some of these topics can be found in chapter 6.3.

7. Epilogue

While writing this thesis I noticed that I made some mistakes in the initial approach to this research. The following epilogue is my opinion on this matter and I hope it might explain some discrepancies that you might have experienced while reading this thesis.

During this research, I often went down the rabbit hole and felt a little bit like Alice in Wonderland. I underestimated how big the field of NLP is and many times felt lost in the vast amount of details of specific subjects (for example all the different types of word embedding and topics such as masked-language modeling and next-sentence prediction).

At the start of this project, I tried to scope the research question as much as possible but in the end, I bit off a bit more than I could chew. To counteract this problem I create a scope in time to prevent the risk of not finishing this research. A large downside to this strategy is that many points in this paper could have been worked out in much more detail (as can be seen in Chapter 6).

One reason that this issue occurred could also have something to do with a mismatch in methodology that I experienced whilst reading papers. In my experience, there is a gap between how research papers are written within the domain of machine learning and papers that are written within the domain of applied science.

Many papers I read are scoped to test 1 specific model on 1 specific dataset. The only thing they report is performance metrics and scores. If you compare to papers in the domain of applied science there is much more attention to the rigor and relevance of the issue at hand.

I feel like I tried to please both worlds in this research. This experience was amplified by the interviews and contact moments with domain experts. Interviews with ML experts were always about ML models, how I used those models, and how I could calculate their performance. But in dialogues with professors and supervisors, the focus was more on the role of feedback and applying these types of AI techniques in an educational domain.

This is the first time I performed research of this size and it has been tremendously educational. Nevertheless, some mistakes were made and there are choices that I would do differently next time.

Sources

Alhami, I. (2011). Automatic Code Homework Grading Based on Concept Extraction. *International Journal of Software Engineering and Its Applications*, *5*(4).

Ali Alvi, & Paresh Kharya. (2021). *Microsoft Research Blog*.

Allamanis, M., & Sutton, C. (2016). A Convolutional Attention Network for Extreme Summarization of Source Code. *Proceedings of Machine Learning Research*, *48*.

Bilal, M., & Almazroi, A. A. (2022). Effectiveness of Fine-tuned BERT Model in Classification of Helpful and Unhelpful Online Customer Reviews. *Electronic Commerce Research*, 1–21. https://doi.org/10.1007/S10660-022-09560-W/FIGURES/3

Bloom, B. S. (1956). *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*.

Brigato, L., & Iocchi, L. (2020). A Close Look at Deep Learning with Small Data. *ICPR* , *3*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Openai, D. A. (2020). *Language Models are Few-Shot Learners*.

Chaudhari, D. D., & Pawar, A. V. (2022). A Systematic Comparison of Machine Learning and NLP Techniques to Unveil Propaganda in Social Media. *Https://Services.Igi-Global.Com/Resolvedoi/Resolve.Aspx?Doi=10.4018/JITR.299384*, *15*(1), 1–14. https://doi.org/10.4018/JITR.299384

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). *BERTje: A Dutch BERT Model*. https://arxiv.org/abs/1912.09582v1

Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERT: a Dutch RoBERTa-based Language Model. *Findings of the Association for Computational Linguistics Findings of ACL: EMNLP 2020*, 3255–3265. https://doi.org/10.48550/arxiv.2001.06286

Delobelle, P., Winters, T., & Berendt, B. (2022). RobBERTje: a Distilled Dutch BERT Model. *Computational Linguistics in the Netherlands Journal*, *11*, 125–140. https://doi.org/10.48550/arxiv.2204.13511

Delobelle, P., Winters, T., Berendt, B., & Leuven, K. (2021). RobBERTje: A Distilled Dutch BERT Model. *Computational Linguistics in the Netherlands Journal*, *11*, 125–140. https://www.clinjournal.org/clinj/article/view/131

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018a). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186. https://arxiv.org/abs/1810.04805v2

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018b). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 4171–4186. https://doi.org/10.48550/arxiv.1810.04805

Edwards, A., Camacho-Collados, J., Ene De Ribaupierre, H., & Preece, A. (2020). Go Simple and Pre-Train on Domain-Specific Corpora: On the Role of Training Data for Text Classification. *Proceedings of the 28th International Conference on Computational Linguistics*, *481*, 5522–5529. https://huggingface.co/ssun32/bert_twitter_turkle#

European Commission, & Directorate-General for Communications Networks. (2019). *Ethics guidelines for trustworthy AI*. https://data.europa.eu/doi/10.2759/346720

Ezen-Can, A. (2020). *A Comparison of LSTM and BERT for Small Corpus*. https://arxiv.org/abs/2009.05451v1

Ferguson, P. (2009). Student perceptions of quality feedback in teacher education. *Http://Dx.Doi.Org/10.1080/02602930903197883*, *36*(1), 51–62. https://doi.org/10.1080/02602930903197883

Forsyth, S., & Mavridis, N. (2021). Short Answer Marking Agent for GCSE Computer Science. *EDUNINE 2021 - 5th IEEE World Engineering Education Conference: The Future of Engineering Education: Current Challenges and Opportunities, Proceedings*. https://doi.org/10.1109/EDUNINE51952.2021.9429163

Gao, S., Alawad, M., Young, M. T., Gounley, J., Schaefferkoetter, N., Yoon, H. J., Wu, X. C., Durbin, E. B., Doherty, J., Stroup, A., Coyle, L., & Tourassi, G. (2021). Limitations of Transformers on Clinical Text Classification. *IEEE Journal of Biomedical and Health Informatics*, *25*(9), 3596–3607. https://doi.org/10.1109/JBHI.2021.3062322

Gedye, S. (2010). Formative assessment and feedback: a review. *Planet*, *23*(1), 40–45. https://doi.org/10.11120/plan.2010.00230040

Gillioz, A., Casas, J., Mugellini, E., & Khaled, O. A. (2020). Overview of the Transformer-based Models for NLP Tasks. *Proceedings of the 2020 Federated Conference on Computer Science and Information Systems, FedCSIS 2020*, 179–183. https://doi.org/10.15439/2020F20

González-Carvajal, S., & Garrido-Merchán, E. C. (2020). *Comparing BERT against traditional machine learning text classification*. https://doi.org/10.48550/arxiv.2005.13012

Hattie, J., & Timperley, H. (2016). The Power of Feedback: *Http://Dx.Doi.Org/10.3102/003465430298487*, *77*(1), 81–112. https://doi.org/10.3102/003465430298487

Hevner, A. (2004). *Design Science in Information Systems Research*. https://www.researchgate.net/publication/201168946_Design_Science_in_Information_Systems_Research

Hevner, A. R. (2007). A Three Cycle View of Design Science Research. *Scandinavian Journal of Information Systems*, *19*(2).

Hift, R. J. (2014). Should essays and other open-ended-Type questions retain a place in written summative assessment in clinical medicine? *BMC Medical Education*, *14*(1), 1–18. https://doi.org/10.1186/S12909-014-0249-2/TABLES/1

Jiao, S., Gao, Y., Feng, J., Lei, T., & Yuan, A. X. (2020). *Does deep learning always outperform simple linear regression in optical imaging?*

Kintu, M. J., Zhu, C., & Kagambe, E. (2017). Blended learning effectiveness: the relationship between student characteristics, design features and outcomes. *International Journal of Educational Technology in Higher Education*, *14*(1), 1–20. https://doi.org/10.1186/S41239-017-0043-4/TABLES/6

Kowsari, K., Meimandi, K. J., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text Classification Algorithms: A Survey. *Information 2019, Vol. 10, Page 150*, *10*(4), 150. https://doi.org/10.3390/INFO10040150

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., & Allen, P. G. (2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach*. https://doi.org/10.48550/arxiv.1907.11692

Mike Wu, Chris Piech, & Chelsea Finn. (2021, July 20). *Meta-Learning Student Feedback to 16,000 Solutions*. http://ai.stanford.edu/blog/prototransformer/#:~:text=The students%27 reception to the,feedback provided by human instructors.

O'gorman, K., & Macintosh, R. (2015). *Research Methods for Business and Management* (Second edition). Goodfellow Publishers Ltd.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep contextualized word representations. *NAACL HLT 2018 - 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, *1*, 2227–2237. https://doi.org/10.48550/arxiv.1802.05365

Polat, M. (2020, October). *(1) (PDF) Analysis of Multiple-choice versus Open-ended Questions in Language Tests According to Different Cognitive Domain Levels*. Novitas-ROYAL . https://www.researchgate.net/publication/357529129_Analysis_of_Multiple-choice_versus_Open-ended_Questions_in_Language_Tests_According_to_Different_Cognitive_Domain_Levels

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language Models are Unsupervised Multitask Learners. *OpenAI Blog* , *1*(8). https://github.com/codelucas/newspaper

Samek, W., Wiegand, T., & Müller, K.-R. (2017). *EXPLAINABLE ARTIFICIAL INTELLIGENCE: UNDERSTANDING, VISUALIZING AND INTERPRETING DEEP LEARNING MODELS*.

Selva Birunda, S., & Kanniga Devi, R. (2021). A review on word embedding techniques for text classification. *Lecture Notes on Data Engineering and Communications Technologies*, *59*, 267–281. https://doi.org/10.1007/978-981-15-9651-3_23/TABLES/1

Shute, V. J. (2008). Focus on formative feedback. *Review of Educational Research*, *78*(1), 153–189. https://doi.org/10.3102/0034654307313795

Si, S., Wang, R., Wosik, J., Zhang, H., Dov, D., Wang, G., Henao, R., & Carin, L. (2020). Students Need More Attention: BERT-based Attention Model for Small Data with Application to Automatic Patient Message Triage. In *Proceedings of Machine Learning Research* (Vol. 126, pp. 436–456). PMLR. https://proceedings.mlr.press/v126/si20a.html

Vabalas, A., Gowen, E., Poliakoff, E., & Casson, A. J. (2019). Machine learning algorithm validation with a limited sample size. *PLOS ONE*, *14*(11), e0224365. https://doi.org/10.1371/JOURNAL.PONE.0224365

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention Is All You Need. *Advances in Neural Information Processing Systems*, *2017-December*, 5999–6009. https://doi.org/10.48550/arxiv.1706.03762

Wang, X., Gulenman, T., Pinkwart, N., De Witt, C., Gloerfeld, C., & Wrede, S. (2020). Automatic assessment of student homework and personalized recommendation. *Proceedings - IEEE 20th International Conference on Advanced Learning Technologies, ICALT 2020*, 150–154. https://doi.org/10.1109/ICALT49669.2020.00051

Wang, Y., Yao, Q., Kwok, J. T., & Ni, L. M. (2020). Generalizing from a Few Examples. *ACM Computing Surveys (CSUR)*, *53*(3). https://doi.org/10.1145/3386252

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., & Wesslén, A. (2000). *Experimentation in Software Engineering*.

Wu, M., Goodman, N., Piech, C., & Finn, C. (2021). *ProtoTransformer: A Meta-Learning Approach to Providing Student Feedback*. 1–19.

Zhang, L., Huang, Y., Yang, X., Yu, S., & Zhuang, F. (2019). An automatic short-answer grading model for semi-open-ended questions. *Interactive Learning Environments*, *0*(0), 1–14. https://doi.org/10.1080/10494820.2019.1648300

Zhang, Y., & Ling, C. (2018). A strategy to apply machine learning to small datasets in materials science. *Npj Computational Materials*, *4*(1). https://doi.org/10.1038/s41524-018-0081-z

Zulqarnain, M., Ghazali, R., Rehan, M., Mazwin, Y., & Hassim, M. (2020). A comparative review on deep learning models for text classification. *Indonesian Journal of Electrical Engineering and Computer Science*, *19*(1), 325–335. https://doi.org/10.11591/ijeecs.v19.i1.pp325-335

Appendix A: All questions in the exam (in Dutch)

QID	Question
0	Een belangrijk model over teamontwikkeling is het model van Tuckman. Volgens Tuckman is het doorlopen van alle fasen in vaste volgorde noodzakelijk om als team uit te groeien tot een optimaal functionerende groep. Benoem de vijf fasen van het model van Tuckman voor teamontwikkeling (5 punten)
1	Afhankelijk van het type projectaanpak kennen we verschillende vormen van faseren. Welke drie soorten werk (achtereenvolgende stappen) komen echter in alle faseringen terug? (5 punten)
2	Licht in maximaal dertig woorden toe wat er wordt bedoeld met 'de duivelsdriehoek' van projecten. (5 punten)
3	A) Noem een voorbeeld van een beheerscyclus. (2 punten)
4	B) We hebben een zestal beheersaspecten onderscheiden. Benoem deze zes aspecten. (3 punten)
5	C) Een van de beheersaspecten kun je in kaart brengen met het RASCI-model of RACI-model. Licht in maximaal 30 woorden toe wat je met dit model kunt doen. (5 punten)'
6	A) Formuleer een projectdoelstelling. (5 punten)
7	B) Formuleer een projectresultaat. (5 punten)
8	C) Formuleer een scope voor dit project op basis van de informatie uit de casus. (5 punten)
9	Je mag voor dit project één idee inzenden. Welke fasering is het meest passend voor een dergelijk project en waarom? (5 punten)'
10	Je besluit om mee te doen en een idee in te zenden. Maak een WBS voor dit project. Het detailniveau moet dermate zijn dat je beste vriend van de opleiding Technische Bedrijfskunde begrijpt wat je voor dit project allemaal moet doen. LET OP! Deze opgave (tekenen van een WBS) dien je op papier te maken en in te leveren bij de surveillant en geef in het tekstvak aan dat je de WBS hebt getekend Zet op iedere ingeleverde pagina bovenaan je naam en klas! (20 punten)
11	Benoem drie externe stakeholders voor je project. Licht per stakeholder in maximaal twee zinnen toe waarom dit een stakeholder voor je project is. (9 punten)
12	Het ministerie vraagt jou om bij het uitrollen van je project een 'Project start up' te leiden. Bedenk minimaal twee zaken welke je in deze bijeenkomst wilt bespreken. (5 punten)
13	A) Benoem een preventieve maatregel voor dit risico. (3 punten)
14	B) Benoem een curatieve maatregel voor dit risico. (3 punten)

Appendix B: Papers used to inventorize used techniques.

Metric	Type of data	Recommended model	Tested models	Year	Name of paper	Reference
F1, Accuracy	Reviews	BERT	BERT VS BOW (K-NN, NB AND SVM)	2022	Effectiveness of Fine-tuned BERT Model in Classification of Helpful and Unhelpful Online Customer Reviews	(Bilal & Almazroi, 2022)
F1, Accuracy	Propoganda social media	SVM	NB, SVM, DT, KNN	2022	A Systematic Comparison of Machine Learning and NLP Techniques to Unveil Propaganda in Social Media	(Chaudhari & Pawar, 2022)
Unclear	Programming assignments	CodeBERT	RoBERTa, codeBERT, PythonBERT, LSTM vs Transofrmers	2021	ProtoTransformer: A Meta-Learning Approach to Providing Student Feedback	(Wu et al., 2021)
F1, Accuracy	Clinical texts (large files)	CNN HISAN	BERT vs CNN or HISAN	2021	Limitations of Transformers on Clinical Text Classification	(Gao et al., 2021)
Accuracy	Short Answers	unclear	BOW, semantic matching (pipeline)	2021	Short Answer Marking Agent for GCSE Computer Science	(Forsyth & Mavridis, 2021)
Accuracy	Student homework	unclear	RF, LSVC, MNB, LR	2020	Automatic assessment of student homework and personalized recommendation	(X. Wang et al., 2020)
F1	Multiple datasets	Fast Tekst	Fast Tekst vs Roberta vs LR (baseline)	2020	Go Simple and Pre-Train on Domain-Specific Corpora: On the Role of Training Data for Text Classification	(Edwards et al., 2020)
Accuracy	Question intent	LSTM on small dataset	LSTM VS BERT	2020	A Comparison of LSTM and BERT for Small Corpus	(Ezen-Can, 2020)
F1, PRECISION, RECALL, AUC	medical docs	LESA-BERT	SVM, CNN, BI-LSTM, BERT, BIO-BERT	2020	Students Need More Attention: BERT-based Attention Model for Small Data with Application to Automatic Patient Message Triage	(Si et al., 2020)
F1, PRECISION, RECALL, accuracy	short ansers networking CS	unclear	SBERT	2020	Automatic Grading System Using Sentence-BERT Network	(Forsyth & Mavridis, 2021)
Accuracy	Multiple datasets including reviews	GRU, LSTM	DBN, CNN, GRU, LSTM	2020	A comparative review on deep learning models for text classification	(Zulqarnain et al., 2020)
Accuracy	IMDB	BERT	BERT vs VC, LR, LSVC, MNB, RC	2020	Comparing BERT against traditional machine learning text classification	(González-Carvajal & Garrido-Merchán, 2020)
Accuracy	Student homework	LSTM	LSTM	2019	An automatic short-answer grading model for semi-open-ended questions	(L. Zhang et al., 2019)