Automated summary scoring using a linguistic feature approach

Christa Meelker
Master Thesis

Supervisors:
Arno Siebes - *Utrecht University*
Remco Feskens - *Cito*
Tjeerd Hans Terpstra - *Cito*

Computer Science
Utrecht University
July 2021

Abstract

Summary-writing tasks are often used to assess reading comprehension of students. Grading these types of tasks is time-consuming and teachers have difficulty being consistent when grading. The goal of this research is therefore to explore and evaluate the possibilities of automating summary grading. Previous research has shown that students with an extensive mental model, and thus a good understanding of the original text, write high-quality summaries. Linguistic features can therefore be used to measure summary quality. A total of 82 different linguistic features is calculated for a dataset of 914 short Dutch summaries. These summaries have been graded by teachers. Through cross-validated feature selection, an optimal set of features is selected for both a regression and classification model. The regression model can be used to predict a grade and has an explained variance of 0.71. The classification model can be used to predict a 'Fail' or 'Pass' label and has an area-under-the-ROC curve of 0.91. It can therefore be concluded that linguistic feature-based models can successfully be used to automate summary grading. The models developed in this research could potentially replace a second or third reader.

Contents

1 Introduction

Summary-writing tasks are a common and effective method to assess reading comprehension of students [1]. Test and exam organizations like Cito, the Dutch national institute for educational measurement, use this type of task along with multiple-choice and open-answer questions to assess the ability of students [2]. The advantage of summary-writing tasks compared to other types of test questions is that they give a good impression of a student's mental representation of a text [1]. The disadvantage of such tasks, however, is that they are time-consuming and thus costly to grade [3]. Moreover, it is extremely difficult for teachers to be consistent when grading summaries [4]. Even though most teachers use specialized rubrics or checklists when grading, the effect of fatigue, bias or ordering can cause a lot of disagreement between teachers [5, 6, 4].

Natural language processing (hereinafter NLP) is a field in machine learning science that combines the techniques of linguistics, computer science, and artificial intelligence to make machines 'understand' written text [7]. Automated essay scoring (hereinafter AES) is an educational application of NLP, which is specialized in grading or classifying essays [8, 9]. AES came to being in the 1960s, when Page first experimented with computational methods to assess text responses [4, 5]. From extensive essays to short answers, AES techniques have been used for a variety of question types [4].

Automatic grading can be cost-effective and can improve consistency [6]. That being said, complete substitution of human graders is often not desirable. Machines are said to lack, for instance, the ability to appreciate creativity and truthfulness in writing [10]. Substituting a second or third reader, however, or combining an automated and human-grade, could be beneficial to capture the best of both worlds [11].

This research is a collaboration between Utrecht University and Cito. The goal of this research is to explore and evaluate the possibilities of automating summary grading. The research question below follows from this domain demand. After the literature study (section 2) and an examination of the available data (section 3), a more technical data science question is formulated in section 4. Throughout this research, ideas were exchanged with fellow master students Bosma en Zoetmulder who worked on a similar research question [12, 13].

Research question:
To what extent can techniques of automated essay grading be used to assess student's summaries?

2 Literature study

2.1 Automated essay grading

AES is a technique that uses NLP to automatically score free-text responses such as essays or summaries [14]. It involves supervised machine learning, which means that a prediction model is trained on data scored by humans [9]. Such a prediction model is considered successful if the predicted grades correlate well with the grades provided by humans [15].

There are three different approaches to supervised learning in the field of AES. The first is to create a regression model, where the goal is to predict an actual grade. The second approach is creating a classification model, where the goal is to label (e.g. 'Fail' or 'Pass') a text. The third approach is to rank essays based on their quality [9].

Many sophisticated and present-day AES systems use neural networks to train a prediction model [9]. An important advantage of neural networks is the fact that there is no need for feature engineering. A disadvantage, however, is that for neural models to perform well, they have to be trained on enormous datasets. This is not a problem when working with English texts, but for other languages the resources are limited. In those cases, the traditional method of feature engineering might be more suitable [9].

2.2 Feature-based methods

In the context of AES, features are measurable characteristics of a text. The mean word or sentences length of a text can, for instance, be considered a simple feature. More sophisticated features can incorporate aspects such as grammar, vocabulary, or tone of a text [14]. When looking at summaries specifically, different types of features can be distinguished.

Firstly, features can be based on the similarity between a model summary and the student's summary. This is a technique that originates from the field of automatic short answer grading. A short answer question typically tests external knowledge instead of reading comprehension. Similarity-based features are, however, also applicable to summaries since graders of summary-writing tasks often base their grade on the similarity to an ideal summary [6, 4]. Methods like ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and BLEU (BiLingual Evaluation Understudy) can be used to calculate the lexical and phrasal overlap between the model answer and the student summary [15, 1]. Both these metrics make use of N-grams, which are sequences of N words in a text [16].

Secondly, features can be based on the similarity between the text that was summarised and the student's summary. This can be measured through, for instance, the amount of N-grams that are directly copied from the original text. BLUE and ROUGE can also be used for this purpose. More features can be derived by taking the ratio of copied N-grams compared to the total amount of

N-grams in a text, or by taking the length of the longest N-gram copied from the original text [1].

Thirdly, features can be solely based on the linguistic characteristics of the student's summary. As mentioned before, summary-writing tasks are an excellent method to assess a student's mental model of a text [1]. Research by Nicolás-Conesa [17] has shown that a sophisticated mental model results in a higher-quality text. Linguistic features have been proven to be good indicators of writing proficiency, and can therefore be helpful in measuring summary quality [18]. So if a student understands the original text and what is being asked of him/her in the summary-writing task, the student will be likely to write a high-quality text.

2.3 Linguistic features

A lot of research has been done into the effectiveness of linguistic features. Vajjalla [14] looked into the role of linguistic features in AES of non-native learners. Zesch et al. [3] looked into task-independent features and Ke and Ng [9] did a state-of-the-art survey of AES methods. Based on the findings of the above-mentioned research the following linguistic feature categories can be defined:

- **Style:** Lexical diversity can be an indicator of good style. It is typically measured by comparing the total amount of tokens in a corpus to the total number of unique tokens in a text [3, 14]. Another indicator of good style is word knowledge, which can be measured by looking at the individual word frequency of a text compared to the complete corpus [3].

- **Word types:** Part-of-speech tags indicate for each token the part of speech and grammatical categories such as tense and number. Features in this category are usually based on the ratio between different POS tags (e.g. noun-verb ratio) [3, 14] .

- **Syntactic Complexity:** Features extracted from parse trees, such as average tree height or the number of subtrees [3, 14].

- **Readability:** A good text is neither too easy nor too hard to read. Metrics such as the Flesch-Kindcaid Reading Ease, which are mostly based on ratios of the word, syllable, and sentence count give an idea of how readable a text is [3, 14, 9].

- **Coherence:** Cohesion and coherence can, for instance, be measured through the amount of connectives (e.g. 'and', 'although', 'however') in a text.[3, 14, 9].

- **Errors:** Spelling and grammar tools can be used to count the number of errors in a text [3, 14].

- **Length:** For instance, the number of words or sentences in a text [3, 14, 9].

Apart from length features, which are known to be good predictors of grades, little is known about which features are most predictive across different datasets [14].

2.4 Overview

An overview of the different types of automated essay scoring methods discussed can be found in figure 1.

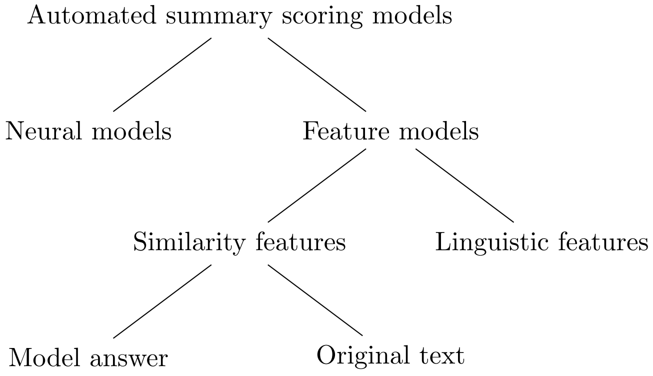


Figure 1: Types of automated summary scoring models

3 Data

3.1 Task description

The dataset consists of answers to a summary-writing task written by students of secondary and tertiary education. The task was part of an education-level reference test, which can be used to determine the ability level of students. There were different versions of the same test, meaning that the order of questions varied across the different versions. This test was low-stakes, meaning that there were no personal consequences such as basing a students school-level advice on their test-performance. The summaries were originally handwritten, but have been transcribed by Cito. The dataset is completely anonymized, owned by Cito, and not publicly available.

Students were asked to summarize an article published by NRC Next, which is a Dutch daily newspaper targeted to the higher educated [19]. Next.checkt is a column of NRC Next, which investigates the validness of statements made by external parties. For this article, the following statement made by Kleenex, a company that sells facial tissues, is being checked: "One out of five people suffer from hay fever and this number is rising drastically". The accompanying summary-writing task had a specific objective and goes as follows (translated from Dutch):

> *Next.checkt concludes that the statement by Kleenix is unfounded. Write a well-flowing summary in no more than 150 words of the counter-arguments on the basis of which the judgment 'unfounded' by next.checkt is justified. Limit yourself to the main arguments.*

The summaries have been graded on a 0-15 scale by human raters. Each summary was graded by one person, using a provided grading rubric. The rubric contained the following guidelines: There we 4 main arguments identified in the original text. For each of the arguments, a maximal amount of points was given, depending on the presence of important details. A maximum of 12 points could be earned for this part. The remaining 3 points were given for the flow and structure of the text. The dataset does not include information about how well students scored on each of the individual items.

3.2 Integration

The dataset was provided as a ready-to-use CSV file. Due to some encoding problems during the creation of this file, special characters were not displayed correctly. The summaries, provided as text files, therefore had to be integrated with the CSV file using ANSI encoding. This integration was done by Bosma [12], more details can be found in her thesis.

3.3 Exploration

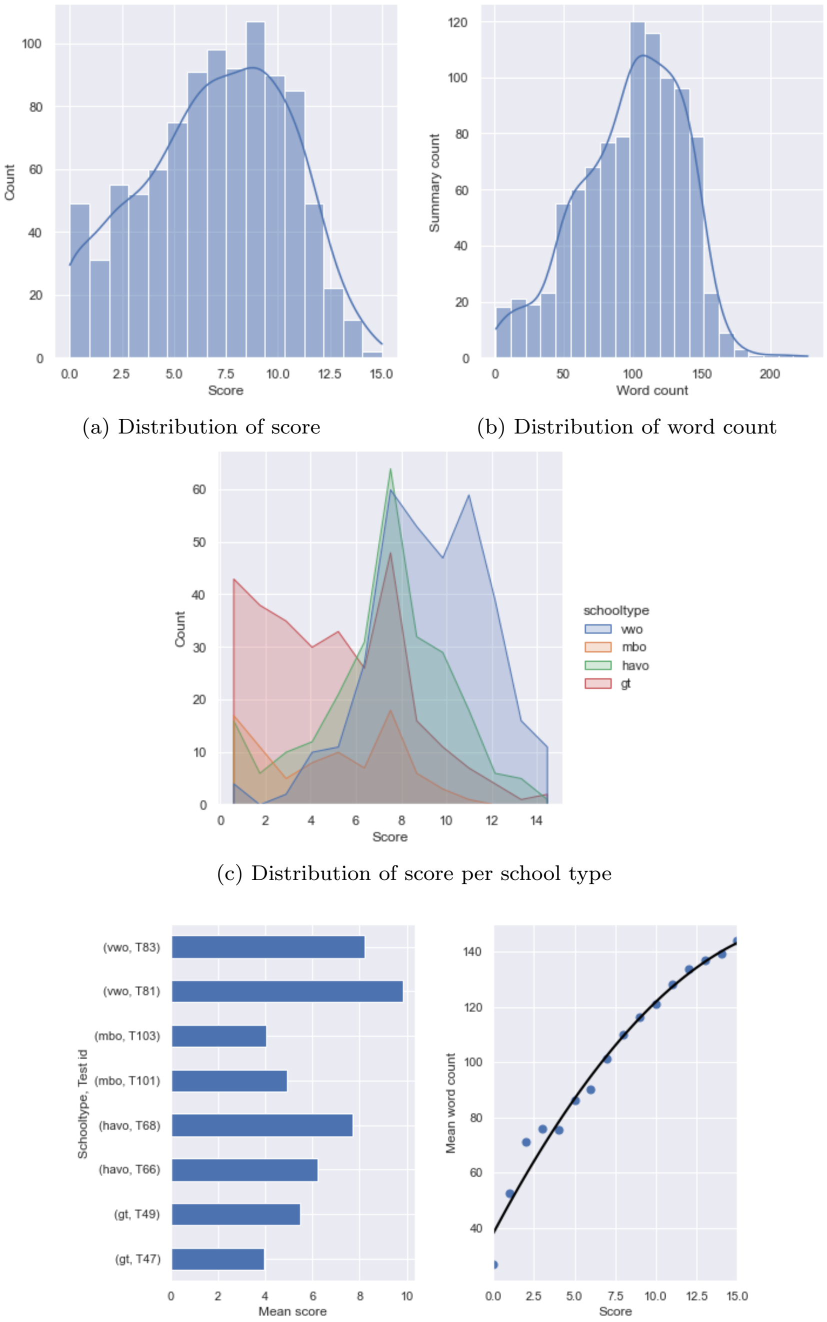
The dataset contains a total of 970 summaries. The distribution of score is somewhat skewed (see figure 2a). What becomes clear from this figure is that a remarkable amount of students score 0 points. The score distribution per school type can be found in figure 2c. Table 1 shows the number of summaries and the mean score per school type. As expected, VWO students score highest, MBO and VMBO-GT lowest.

School type	Number of summaries	Mean score
MBO	86	4.61
VMBO-GT	294	4.81
HAVO	251	7.13
VWO	339	9.29

Table 1: Statistics per school type

As mentioned before, the summary-writing task appeared at different locations in different versions of the same test. There were 8 different versions in total, of which 2 per school type. Students from 49 different schools made this test. Figure 2d shows the mean score per test. There is a difference in score between the different test-school type pairs, which could be explained by student motivation or concentration: If the task appeared at the end of the test this could result in a lower score and vice versa. These are only speculations since information about the question order is not available.

The word count distribution, which is also skewed, can be found in figure 2b. It is remarkable that even though the word limit was 150, a substantial part of the summaries counts less than 50 or more than 150 words. Figure 2e shows the mean amount of words used per grade. Clearly, there is a clear positive relation between word count and score.



(a) Distribution of score

(b) Distribution of word count

(c) Distribution of score per school type

(d) Mean score per school type and test

(e) Mean word count per score

Figure 2: Data exploration

3.4 Cleaning

3.4.1 Missing values

There were no empty cells in the dataset. There were, however, cases where the transcriber had problems reading the written text. Different transcribers used different methods to document this. Some, for instance, wrote down 'unreadable' whereas others wrote 'unrecognizable'. All of the different cases have been detected and coded to 'unreadable' so the number of unreadable words can later be used as a feature. There were 6 summaries that were not readable at all, those have been removed from the dataset. 70 summaries, which is more than 7% of the dataset, contained one or more unreadable words. The problem with unreadable words is that it is unsure whether those words were only unreadable for the transcriber, or that they were unreadable for the person who graded the summary as well. In the latter case, the summary should be included in the dataset because the typed text then is a valid representation of the written text and matches the grade. If only the transcriber had problems reading the text, the typed version is not identical to the graded version and could be polluting the dataset. The safest option would be to remove all 70 summaries, but since the dataset is already quite small this would have a big impact. It was therefore decided to compromise on this issue and remove summaries with more than 1 unreadable word (21 in total). In those cases, there were often complete phrases missing instead of just one word.

3.4.2 Outliers

An outlier is a value in a dataset that differs significantly from the rest of the values in a dataset. Outliers are often caused by errors in the data collection. If they are not removed from a predictive model they might cause incorrect results [20].

Firstly, during the exploration phase, it was noted that the dataset contained a lot of instances with grade 0 (see figure 2a). Due to the relatively small amount of summaries, all of those have been inspected manually. Some students clearly did not take the task seriously, probably because it was a low-stakes test, and wrote down something along the lines of: "I don't get this". A human grader would always give such a summary 0 points. A machine, however, might not be able the recognize the silliness of this answer and could score it higher than 0. In order to mitigate this risk, those cases have been removed so that they will not infer with the model (11 summaries in total).

Secondly, outliers were detected based on the score and word count relation using the commonly used interquartile range (hereinafter IQR) technique. The IQR is the difference between the first quartile (Q1) and the third quartile (Q3) and therefore contains the middle 50% of all observations in a category [21]. Figure 3 shows the distribution of word count per grade, where the colored boxes show the IQR of word count for each grade. If an observation is lower than Q1 - 1.5 * IQR or higher than Q3 + 1.5 * IQR it is considered an outlier. For instance, a summary with less than 10 words but with a score higher than

10 must be a mistake. In figure 3 observations that fall outside of the whiskers are outliers. Using this method, summaries with a relatively high or low word count were removed (18 summaries in total).

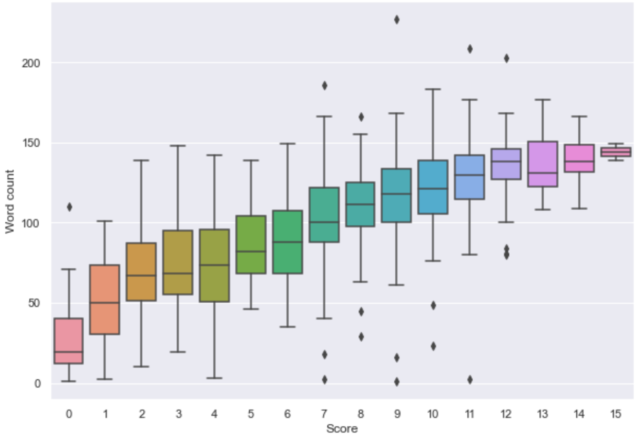
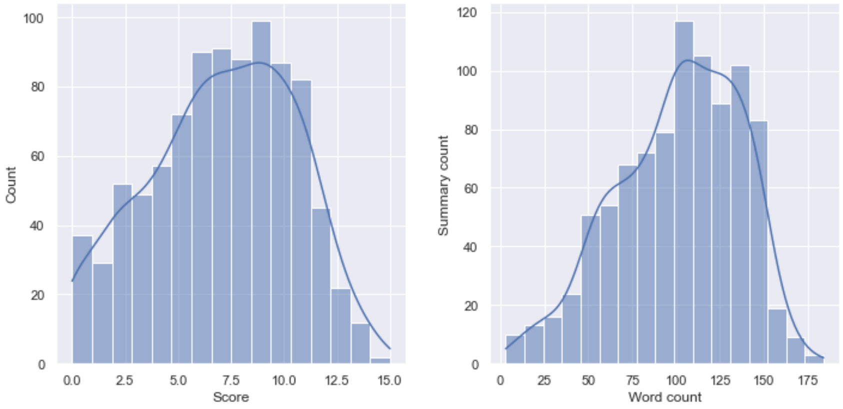


Figure 3: Distribution of word count per score

After removing missing values and outliers there were 914 summaries left. The distribution of score and word count is closer to normal after the cleaning and can be found in figure 4.



(a) Distribution of score

(b) Distribution of word count

Figure 4: Distributions after cleaning

3.5 Preprocessing

When analyzing natural language, text has to be converted into well-defined and linguistic units such as words or sentences [22]. These units can then be for feature extraction. Due to the inherent ambiguities and varieties of human language, preprocessing is extremely sensitive to errors. Figure 5 shows the processing steps that have been carried out. In table 2 the different steps are shown for an example sentence from the dataset.

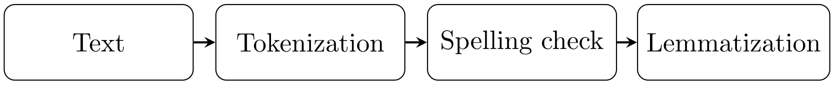


Figure 5: Preprocessing steps

Preprocessing step	Result
Text	Het CMR bewaard sinds 1985 de gediagnostiseerde hooikoorstgevallen waarbij in de periode 1985-2006 een stijging zichbaar is.
Tokenization	het, cmr, bewaard, sinds, 1985, de, gediagnostiseerde, hooikoorstgevallen, waarbij, in, de, periode, 1985-2006, een, stijging, zichbaar, is
Spelling check	het, **crm**, bewaard, sinds, 1985, de, **gediagnosticeerde**, **hooikoorstgevallen**, waarbij, in, de, periode, 1985-2006, een, stijging, zichtbaar, is
Lemmatization	crm, bewaren, 1985, diagnosticeren, hooikoorstgevallen, waarbij, periode, 1985-2006, stijging, zichtbaar

Table 2: Preprocessing example

Firstly, the summaries were tokenized using the spaCy tokenizer [23], which means that all sentences are split into words. In this step, all punctuation is removed from the text and words are set to lower case.

Secondly, misspelled words were corrected (if possible). This was done because of the following reason: If a student wrote down a misspelled version of *hooikoorst* (e.g. *hooikoorst*), a human grader would still understand what the student meant. A machine, however, will not recognize this word, nor its linguistic characteristics. By correcting misspelled words this problem is solved. The spelling corrections were done using a Dutch dictionary published by the foundation OpenTaal [24]. If a word does not exist in the dictionary, but there exists a word that is very similar, the token was replaced with the correctly spelled token. If there was no similar word in the dictionary (e.g. *hooikoorstgevallen*), the token was included in its original form. This is done to make sure that proper names or abbreviations are not excluded from the analysis. For each summary, the total number of corrected words was added as a feature. The spelling correction was, however, not completely flawless. In the example, *CMR*, which is the abbreviation of a dutch organization, is wrongly corrected to *crm.* The advantages of the spelling checker do however outweigh the disadvantages.

Thirdly, the tokens were lemmatized, and stop words were removed using spaCy [23]. Lemmatization means that different forms of one word, such as *walk, walked, walks* are all grouped into its dictionary form *walking.* This is done to make sure that these terms can be analyzed as a single item. The advantage of stop word removal is that it increases the percentage of meaningful tokens [22].

4 Data science question

As discussed in section 3, the available dataset consists of less than 1000 short Dutch summaries with a specific objective. No research in the AES field on this specific type of question was found. Still, the methods discussed in section 2 are applicable to the current research.

The effectiveness of a neural model on this dataset is disputable, as there is the risk of a training set that is too small. Besides, a neural model works like a black box, meaning that it can give limited insights into what exactly makes a good summary. A feature-based model is therefore more appropriate. Since there is no model answer or 'ideal' summary available, and the original text is summarized with an objective, it would be interesting to look into a linguistic feature model. The question then remains whether linguistic features contain enough information to assess the quality of a summary, and if so, which features are most predictive. There is no interest in ranking summaries, so the focus will be on classification and regression models.

Data science question:
To what extent can linguistic features be used to build predictive models to automatically score student's summaries?

Subquestion 1:
Which features are most predictive in such models?

Subquestion 2:
How do the performances of a classifier and a regression model differ?

5 Method

A schematic overview of the model development pipeline can be found in figure 6. Each of the steps will be discussed and motivated in detail in the sections below. The resulting models will be evaluated in section 6. All coding for this research is done in Python. The required packages and versions can be found in appendix A.1.

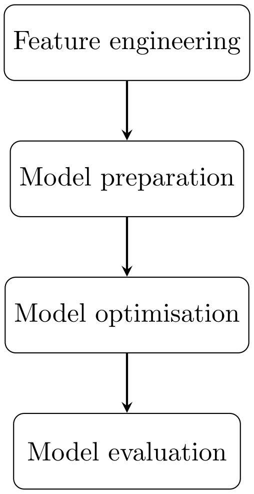


Figure 6: Model development pipeline

5.1 Features

Except for features having to do with document length, little is known about which linguistic features are the most useful predictors of score across different datasets [14, 3]. To find out which features work best for this summary-writing task, different features from almost all categories mentioned in section 2.3 have been created. Some of the features are based on previous research, others are completely new. The category *syntactic complexity* was out of the scope for this research.

- Style
  - **Type token ratio:** $num.types/\sqrt{num.tokens}$. *Num.types* is the amount of different word types in a summary and *num.tokens* is the total amount of tokens in the summary. The same word types defined in the POS tags features were used. Different versions of this feature were found in previous research. This exact version is named root type token ratio by Vajjala and Lu [14, 25]. Yamamoto et al. created a similar feature called lexical richness [5].

  - **Stop word count:** During the preprocessing phase stop words were removed using spaCy [23]. For each summary, the total amount of stop words that was removed was added as a feature.

- Word types
  The POS-tagging function of spaCy [23] was used to detect the following word types: Adjectives, coordinating conjunctions, subordinating conjunctions, symbols, nouns, adpositions, adverbs, numerals, pronouns, interjections, determiners, auxiliary verbs, proper nouns and verbs. For each type, the following features were created:

  - **(word type) Count:** The total amount of words of the specific word type. This features is based on features by Vajalla [14].
  - **(word type) Ratio:** Because the summaries differ a lot in length, this features calculates $word.type.count/word.count$.
  - **(word type) Nr unique:** The amount of unique words of a specific type.
  - **(word type) Variation:** $T.word.type/\sqrt{2 * N.word.type}$. For instance T.verb is the total amount of unique verbs in the corpus. N.verb is the amount of unique verbs in the summary. This features says something about how varied the word use is in a summary, and is based on the corrected verb variation by Lu [25].
  - **Noun to pronoun ratio:** $Noun.count/Pronoun.count$, based on the feature by Vajjala [14].

- Readability
  The python packages Readability [26] has the functionality to implement a set of traditional readability measures for Dutch texts. The following measures are included as features:

  - **Kincaid:** Based on average sentence length and the average number of syllables per word [27].
  - **ARI:** Based on character count and word and sentence length [26].
  - **Coleman Liau:** Relies on characters count per word [26].
  - **Flesch Reading Ease:** Based on average sentence length and the average number of syllables per word [27].
  - **Gunning Fox Index:** Based on average sentence length and the amount of complex words. Complex words in this case are words containing more than two syllables [27].
  - **LIX:** Based on number of words, periods and long words [26].
  - **SMOG Index:** Based on the number of sentences and polysyllables [27].
  - **RIX:** Based on the amount of long words and sentences length [26].

  - **Dale Chall Index:** Based on the amount of difficult words, total word counts and sentence length [26].

- Errors
  - **Error count:** The total amount of spelling errors as described in section 3.5
  - **Unreadable word count**: The amount of words that were unreadable for the transcriber (see section 3.4.1).

- Length
  - **Mean token length:** The mean length of all tokens in a summary.
  - **Mean token length top 60:** This feature is based on the idea that the use of longer words might have a positive relation with the score. This feature calculates the mean length of the 60 longest tokens in a summary.
  - **Mean token length bottom 60:** The mean length of the 60 shortest tokens in a summary.
  - **Sentence count:** The total amount of sentences in a summary, calculated using spaCy [23][14].
  - **Mean sentence length:** The mean amount of tokens per sentence.
  - **Mean sentence length top 4:** The mean sentence length of the 4 longest sentences.
  - **Mean sentence length bottom 4:** The mean sentence length of the 4 shortest sentences.

Even though the coherence category is not explicitly mentioned in the above list, it can be measured through the POS features of the conjunction word types.

A total of 82 features was calculated. For each feature, the mutual information score was calculated with respect to the summary score. The mutual information score expresses how much information about the summary score can be obtained from the feature. The 15 features with the highest scores are listed in table 3.

Rank	Feature	Mutual information
1	GunningFogIndex	0.439
2	RIX	0.423
3	ARI	0.422
4	Kincaid	0.422
5	Stop word count	0.421
6	FleschReadingEase	0.406
7	LIX	0.398
8	Mean token length top 60	0.397
9	Word count	0.358
10	NOUN count	0.343
11	Type token ratio	0.339
12	DaleChallIndex	0.328
13	VERB variation	0.326
14	VERB count	0.316
15	SMOGIndex	0.306

Table 3: Features with highest Mutual Information score

5.2 Models

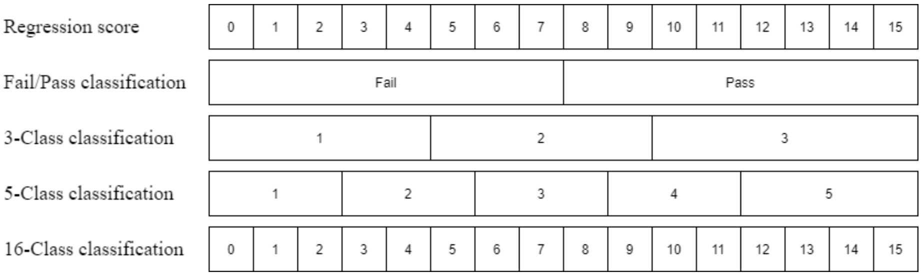
Since this is mainly exploratory research, a regression and multiple classification models will be calculated and evaluated.

5.2.1 Regression

A Ridge regression model is trained using the Scikit-Learn library [28]. Ridge regression is suitable for data with a set of features that may suffer from multicollinearity [29]. This is the case, as a lot of the readability features, for instance, use word and sentence count to calculate readability. It is therefore likely that these features have a linear relation. A linear relation is also expected between features such as word count and sentence count.

5.2.2 Classifiers

For the classification model, the main interest is in a fail-pass classifier. To get more insights into which scores are most difficult to classify, a 3-class, 5-class, and 16-class classifier will also be trained. The 16-class classifier is an alternative to the regression model which makes it possible to compare the performance of both. The division of scores per class can be found in figure 7. Because there are relatively few summaries with a score of 15 and the scores could not be split up equally, the division is made this way.



<table>
<tr><td>Regression score</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td>Fail/Pass classification</td><td colspan="8">Fail</td><td colspan="8">Pass</td></tr>
<tr><td>3-Class classification</td><td colspan="5">1</td><td colspan="5">2</td><td colspan="6">3</td></tr>
<tr><td>5-Class classification</td><td colspan="3">1</td><td colspan="3">2</td><td colspan="3">3</td><td colspan="3">4</td><td colspan="4">5</td></tr>
<tr><td>16-Class classification</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
</table>

Figure 7: The different classes

For all four classifiers, a random forest classifier is trained using the Scikit-Learn library [28]. Random forests are a popular type of estimator that generally perform well [29].

5.2.3 Preparation

A few steps were executed to prepare the features for the models. Firstly, the dataset was split into a train (80%) and test (20%) set. It is important to do this before any of the other model preparations to make sure that the test set is completely untouched and the model is not over-fitted.

Secondly, the lemmas are transformed into a term frequency-inverse document frequency (TF-IDF) vector space. TF-IDF measures the originality of a word based on the number of times a word appears in a text, compared to the number of times a word appears in a corpus. Words that are frequent in a corpus are of less value for an individual text. If, for instance, a specific word is common in good summaries and uncommon in bad summaries, the presence or absence of this word is a good predictor of the quality of a summary. Creating a TF-IDF vector space is crucial because it transforms words, which are hard to interpret for machine learning algorithms, into numerical statistics. The vector space is fitted on the train set, after which both the train en test set are transformed. The lemmas are now represented as sparse matrices. Truncated singular value decomposition is used to decompose the data into more dense matrices that are easier to process. The decomposer is fitted in the train set after which both the train and test set are transformed [28].

Thirdly, all of the features are standardized. Standardization is important because machine learning estimators perform best on normally distributed data [28]. Again, the standardizer is fitted on the train set after which both the train and test set are transformed.

5.2.4 Optimisation

The regression model will be evaluated based on the explained variance. For the classifiers, the area under the ROC curve is used. Both measures were picked because they are common within Cito. Two methods will be used to optimize the models in order to obtain the highest score possible.

Firstly, the most predictive set of features is selected through recursive feature elimination. This means that first a model is trained using the complete set of features. Features that contribute little or not at all to the model are removed from the set. The latter happens recursively until the optimal set of features is selected. A small set of features reduces the training time of a model and makes it less complex and thus easier to interpret [28].

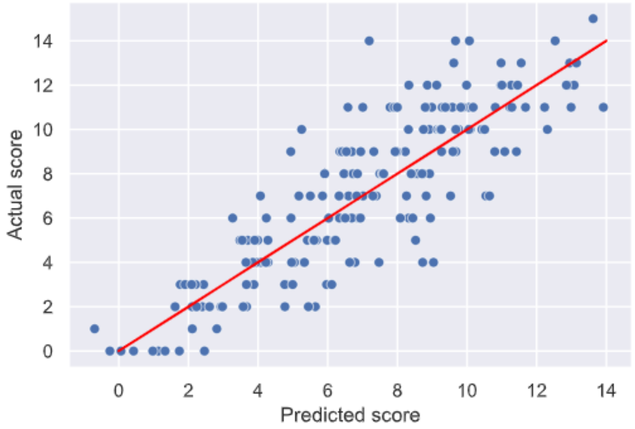
Secondly, the best set of parameters is selected through a grid search. Parameters are the settings for a model which can have a big influence on the accuracy [28]. A range of possible parameter combinations is tried after which the most successful set is selected.

For both the feature selection and parameter tuning a 5-folds cross-validation is used. This means that the training set is split into 5 subsets. Each subset acts as validation set when the other 4 subsets are used for training. The average score for all 5 folds is then taken as the final score. Cross-validation results in a more robust and reliable score [28].

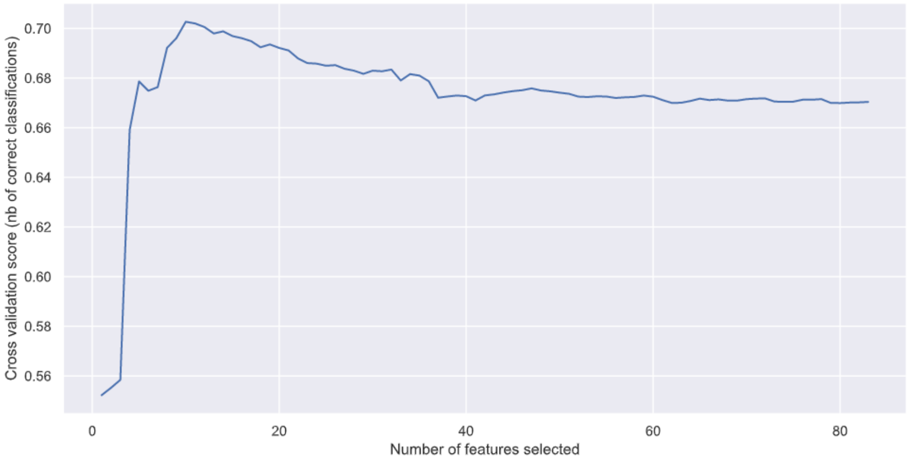
6 Results

6.1 Regression

The regression model has an explained variance of 0.71. A plot of the predicted versus the actual scores can be found in figure 8a. The plot visualizes the inaccuracies of the model. For instance, when looking at an actual score of 7, the predictions can vary from 4 to 9. The plot also shows that the model becomes less accurate for higher scores. When looking at an actual score of 11, the predictions vary from 6.5 to 15. The mean absolute error of this model is 1.46. This means that, on average, an actual score of 7 will be predicted between 5.54 and 8.46.



(a) Predicted vs actual values



(b) Feature selection cross validation scores

Figure 8: Regression results

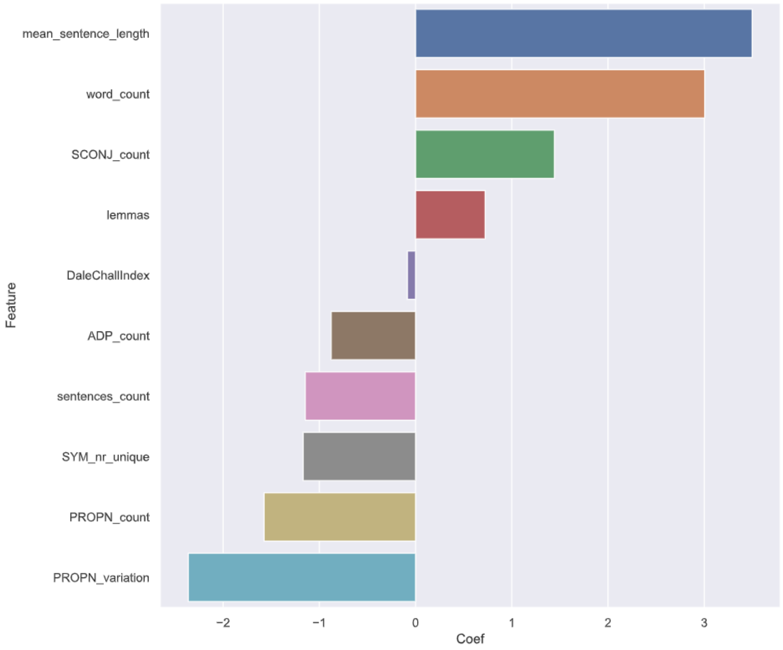


Figure 9: Feature importance regression model

Through cross-validation an optimal number of ten features is selected (see figure 8b). The ten features and their respective regression coefficients are shown in figure 9. The coefficients represent the relation between the feature and score when all other nine features remain constant. For instance, if the mean sentence length increases but the amount of words remain constant the final score will be higher. Also, if the amount of subordinating conjunctions (e.g. although, because, whenever) increases and the other features remain constant, the final score will be higher. The same goes for word count. The coefficient of the lemmas feature is difficult to interpret, as it consists of the TF-IDF matrices. It is clear, however, that these matrices play an important role when predicting a score.

The features with a negative coefficient work the other way around. Students who use more proper nouns (e.g. Finland, Netherlands) while features such as sentences and word count remain constant will score lower. The same goes for proper noun variation, the number of unique symbols, and the number of adpositions (e.g. on, in, by). Similarly, if the number of sentences increases while the amount of words and all other features remain constant the score will be lower. The student is then writing shorter sentences which could explain the lower score.

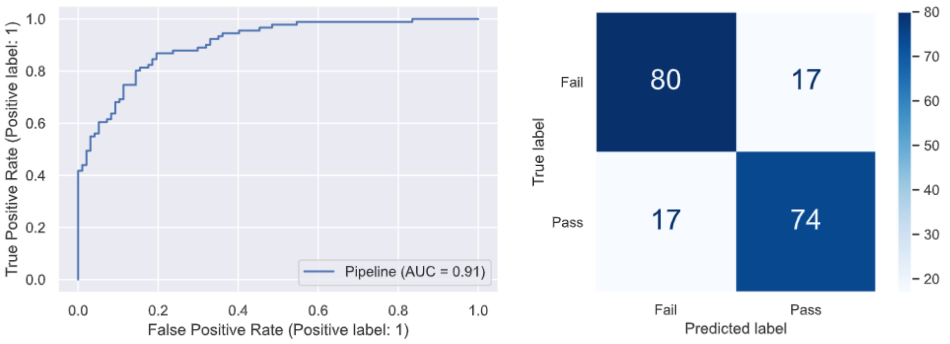
An unexpected finding is the negative coefficient of the DaleChallIndex readability feature. Even though the coefficient is small, the current finding is that a more readable text results in a lower grade. This is in contrast with previous research on readability and text quality [3, 9, 14]. The DaleChallIndex is based on the ratio of difficult words. A word is considered difficult if it is not among the 3000 most frequent words in a 500-million-word Dutch reference corpus [26]. For this summary-writing task, the use of complex words is not beneficial for a higher grade.

6.2 Classifiers

6.2.1 Fail/Pass classifier

The fail/pass classification model has an AUC of 0.91, and an accuracy of 0.82 (see figure 10a). Figure 10b shows the confusion matrix for this model. It becomes clear from this figure that 17 summaries were incorrectly labeled as 'Fail' and 17 summaries were incorrectly labeled as 'Pass'. The model, therefore, does not have a bias in a specific direction.

Through cross-validation an optimal number of 23 features was selected (see figure 11). These 23 features and their weight in the model can be found in figure 12. 7 of these features, which are all in the top 8 of highest weight, were also selected by the regression model. The length features (word count, sentences count, and mean sentence length) are important, as well as the different word type features. Furthermore, the model has included 2 readability measures and 2 style measures.



(a) Area under the ROC curve

(b) Confusion matrix

Figure 10: Fail/pass classifier results

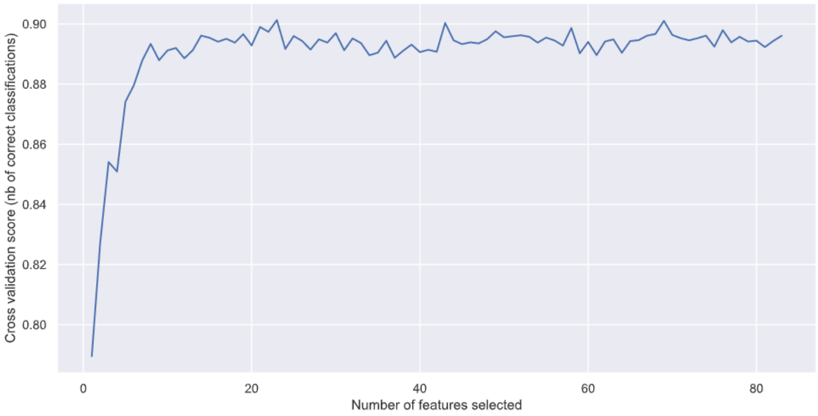


Figure 11: Feature selection cross validation scores

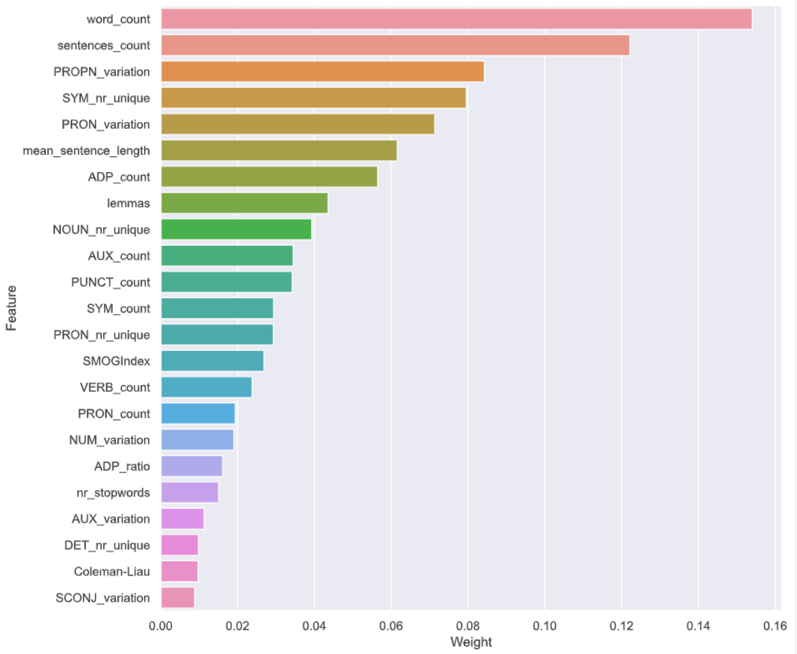
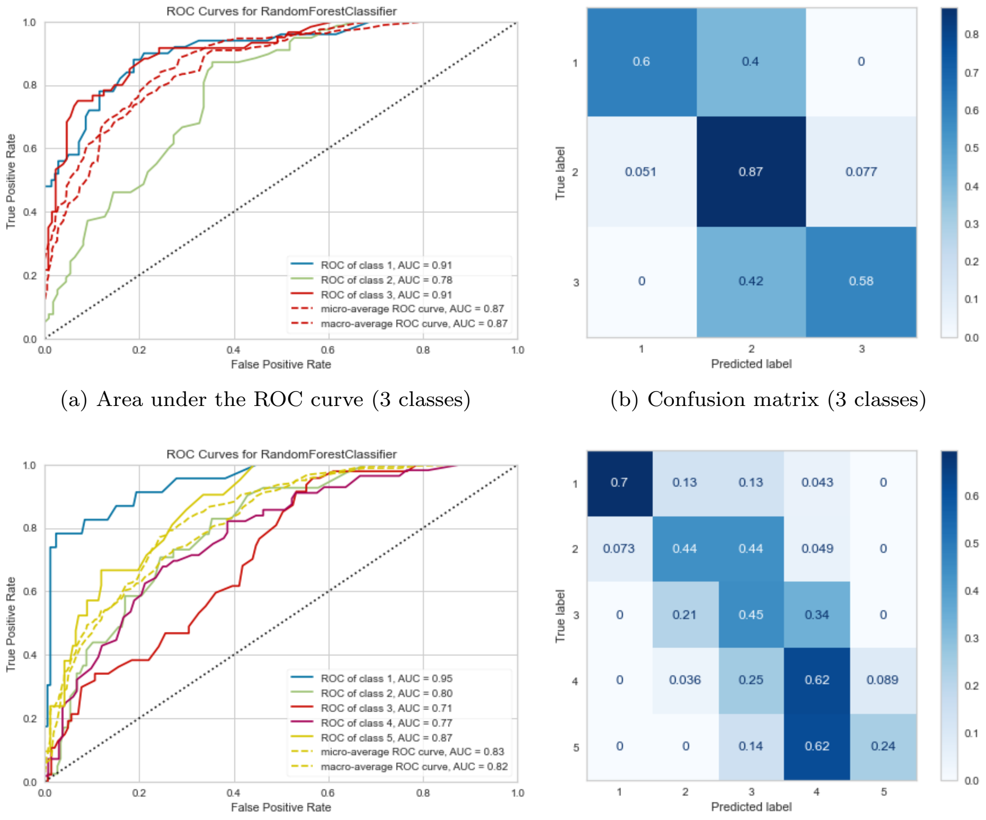


Figure 12: Feature importance fail/pass classifier

6.2.2 Multiclass classifiers

In order to get some insight into the difficulties of a classifier for this dataset, a 3-class and 5-class model were calculated. Figure 13 shows the ROC curves and the confusion matrices for these models. The 3-class model has a mean AUC of 0.87 and an accuracy of 0.70. The results of the 3-class model show that the biggest difficulty lies with class 2, which are the summaries with a score between 5 and 9. This is not unexpected, since exceptionally good or extremely bad summaries are the easiest to score for humans as well.

The 5-class model has a mean AUC of 0.83 and an accuracy of 0.48. The results of the 5-class model show that the model is most successful in classifying low-scoring summaries. In most incorrect cases the model is classifying a summary just one class higher or lower than the actual class. From the confusion matrix, it can be concluded that the model has problems classifying summaries in class 5. This could be explained by the small number of summaries with grade 14 or 15 in the test set.



(a) Area under the ROC curve (3 classes)

(b) Confusion matrix (3 classes)

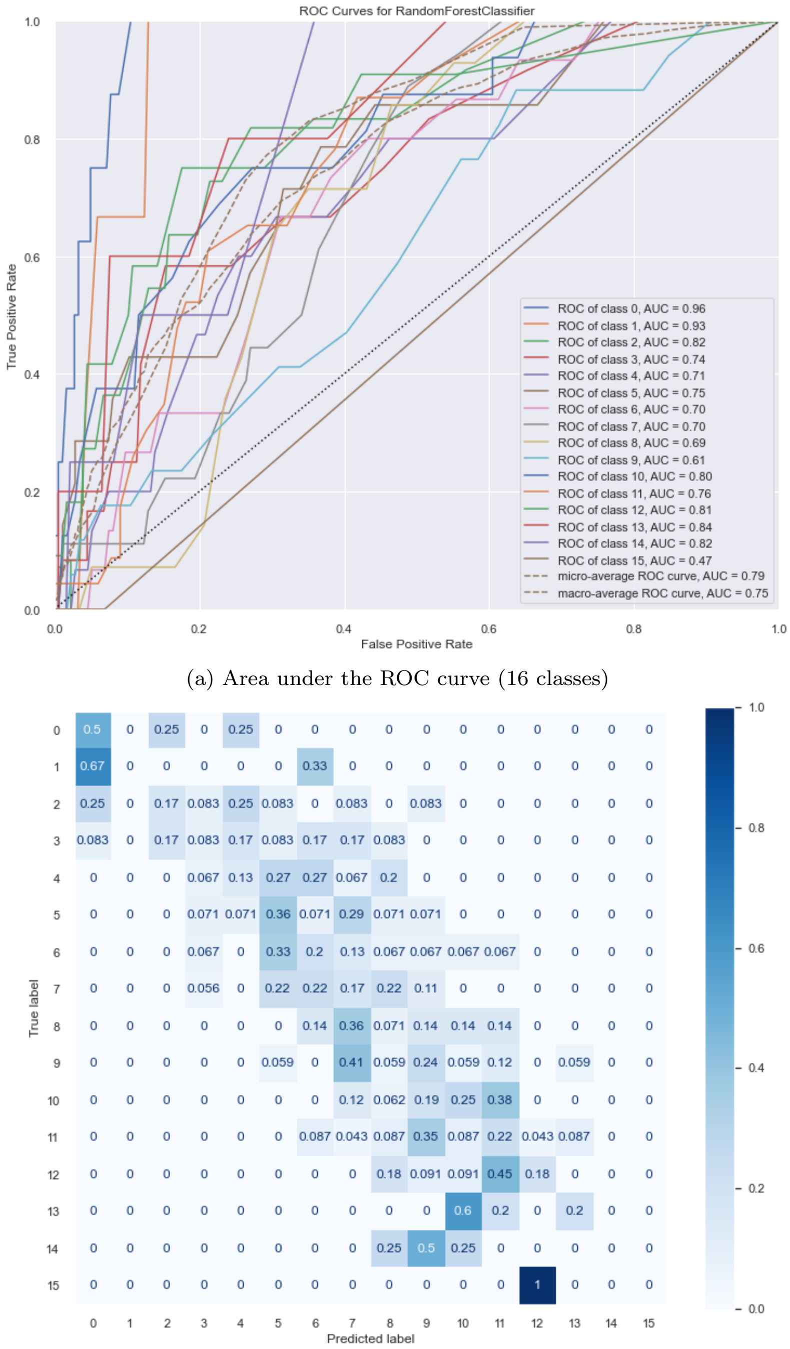
(c) Area under the ROC curve (5 classes)

(d) Confusion matrix (5 classes)

Figure 13: Classification results for 3 and 5 classes

Lastly, a 16-class classifier was trained. This model has an average AUC of 0.79 and an accuracy of 0.20. The explained variance of this model is 0.61. This metric is not meant to be used for classifiers, but since this model is so similar to the regression model the metric is used to compare the two. The corresponding ROC curve can be found in figure 14a. This model has the most difficulty classifying summaries with grades 8, 9 and 15.

The confusion matrix can be found figure 14b. This classifier has a mean absolute error of 2. This means that a summary with grade 7 was on average classified as any grade between 5 and 9. The poor performance of this model could be explained by the small amount of data per class.



(a) Area under the ROC curve (16 classes)

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	0.5	0	0.25	0	0.25	0	0	0	0	0	0	0	0	0	0	0
1	0.67	0	0	0	0	0	0.33	0	0	0	0	0	0	0	0	0
2	0.25	0	0.17	0.083	0.25	0.083	0	0.083	0	0.083	0	0	0	0	0	0
3	0.083	0	0.17	0.083	0.17	0.083	0.17	0.17	0.083	0	0	0	0	0	0	0
4	0	0	0	0.067	0.13	0.27	0.27	0.067	0.2	0	0	0	0	0	0	0
5	0	0	0	0.071	0.071	0.36	0.071	0.29	0.071	0.071	0	0	0	0	0	0
6	0	0	0	0.067	0	0.33	0.2	0.13	0.067	0.067	0.067	0.067	0	0	0	0
7	0	0	0	0.056	0	0.22	0.22	0.17	0.22	0.11	0	0	0	0	0	0
8	0	0	0	0	0	0	0.14	0.36	0.071	0.14	0.14	0.14	0	0	0	0
9	0	0	0	0	0	0.059	0	0.41	0.059	0.24	0.059	0.12	0	0.059	0	0
10	0	0	0	0	0	0	0	0.12	0.062	0.19	0.25	0.38	0	0	0	0
11	0	0	0	0	0	0	0.087	0.043	0.087	0.35	0.087	0.22	0.043	0.087	0	0
12	0	0	0	0	0	0	0	0	0.18	0.091	0.091	0.45	0.18	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0.6	0.2	0	0.2	0	0
14	0	0	0	0	0	0	0	0	0.25	0.5	0.25	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

(b) Confusion matrix (16 classes)

Figure 14: Classification results for 16 classes

7 Conclusion

The goal of this research was to explore and evaluate the possibilities of automating summary grading. From the literature study in section 2 it was concluded that a linguistic feature-based approach is most appropriate for the available dataset. Linguistic features have nothing to do with a summary-writing task, nor do they have any relation with a model or 'ideal' summary. They are, however, good indicators of writing quality. If a student does not understand the original text, nor what is being asked of him in the summary-writing task, the student will not be able to write a high-quality text. Using linguistic features only, a regression model can be used to predict the summary score. The model has an explained variance of 0.71 and a mean absolute error of 1.46. A classifier can be used to label a summary as 'Fail' or 'Pass' with an AUC of 0.91 and an accuracy of 0.82. Both models are most successful in predicting the score or label of low-graded summaries and have the most difficulty predicting high scores.

For both the regression model and the fail/pass classifier the best set of predicting features was selected. Figure 15 shows the percentage of features per category, for both models. In agreement with previous studies, features concerning length are robust predictors of grade. Different word type features, however, also play a big role in predicting a score. Features having to do with style, readability and coherence are also important. Interestingly enough, the number of errors made by students was not selected as a feature by both models.

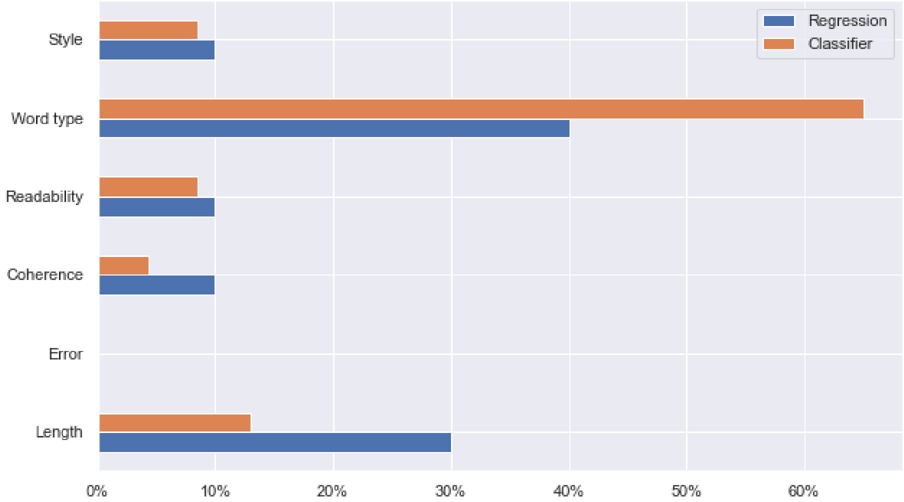


Figure 15: Feature importance per model

To answer the research question 'To what extent can techniques of automated essay grading be used to assess student's summaries?': both a regression and classification model can successfully predict a student's score, especially considering the fact that human graders do usually not agree on scores themselves [5, 6, 4]. The prediction models created in this research project could therefore be used as a second or third reader in future grading.

8 Discussion

8.1 Ethical considerations

An often discussed topic during this research was whether or not to include school type as a feature. As was shown in table 1, the mean score differs a lot across the different school types. MBO students score on average 4.68 points lower. A possible effect of including school type in the model is that two students (one from VWO and one from MBO) writing the exact same summary will score differently. Just because a student is of school type VWO and VWO students score higher on average. On the other hand, MBO students will have more difficulty writing a good summary compared to VWO students. When an MBO student writes a high-quality summary this is much more impressive than when a VWO student writes a high-quality summary. For this reason, it might be desirable if a prediction model is more lenient when grading a summary written by a MBO student. Because the current task was part of a reference test, which can be used to assess a student's and test's school type, it was decided to exclude school type from the model.

8.2 Limitations

The fact that both models are most successful in predicting low grades and have more difficulty predicting high grades can be explained by the amount of data per grade. As shown in figure 4a, there are a lot more summaries with a low grade than summaries with a high grade. Besides, during the development of the models, it was noted that the results are highly dependent on the train-test split. This dependency is also caused by the size of the dataset. A larger dataset would most likely improve the results.

The features that have been selected in this research are proven to be good predictors for this specific summary. There is no guarantee that these features will also work on different summary-writing tasks. A multi-corpus study or a cross-corpus study could improve the generalizability of the models [14].

Another weakness of this dataset is the fact that each summary is graded by one teacher and not all summaries were graded by the same teacher. Since teachers often don't agree on grades, it would be beneficial to have a second-grader per summary [5, 6, 4]. It might be the case that some teachers have been more lenient than others when grading. Since the dataset does not include any information of who graded which summary, this information is unknown. Comparing the predicted grade to a mean human-grade would therefore result in a more robust accuracy.

8.3 Future research

Due to the scope of this research, there were no syntactic complexity features included. Extracting features from clauses and T-units, for instance, might improve the accuracy of the models [30].

Lastly, the findings by Bosma [12] and Zoetmulder [13] could be incorporated to improve results. Bosma did research into similarity features. The most predictive features could be added to the current feature set [12]. Zoetmulder did research into neural models [13]. As mentioned by Ke, the use of feature-based models and neural models should be complementary [9].

References

[1] N. Madnani, J. Burstein, J. Sabatini, and T. O'Reilly, "Automated Scoring of a Summary-Writing Task Designed to Measure Reading Comprehension," *Proceedings of the Eighth Workshop on Innovative Use of NLP for Building Educational Applications*, pp. 163–168, 2013. [Online]. Available: http://www.aclweb.org/anthology/W13-1722

[2] Wikipedia, "Centraal Instituut voor Toetsontwikkeling — Wikipedia, the free encyclopedia," 2021, [Online; accessed 15-June-2021].

[3] T. Zesch, M. Wojatzki, and D. Scholten-Akoun, "Task-Independent Features for Automated Essay Grading," pp. 224–232, 2015.

[4] S. Burrows, I. Gurevych, B. Stein, S. Burrows, I. Gurevych, ·. I. Gurevych, and B. Stein, "The Eras and Trends of Automatic Short Answer Grading," *Int J Artif Intell Educ*, vol. 25, pp. 60–117, 2015.

[5] M. Yamamoto, N. Umemura, and H. Kawano, "Automated essay scoring system based on rubric," *Studies in Computational Intelligence*, vol. 727, pp. 177–190, 2018. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-64051-8_11

[6] N. Süzen, A. N. Gorban, J. Levesley, and E. M. Mirkes, "Automatic short answer grading and feedback using text mining methods," in *Procedia Computer Science*, vol. 169. Elsevier B.V., jan 2020, pp. 726–743.

[7] Wikipedia, "Natural language processing — Wikipedia, the free encyclopedia," 2021, [Online; accessed 15-June-2021].

[8] ——, "Automated essay scoring — Wikipedia, the free encyclopedia," 2021, [Online; accessed 15-June-2021].

[9] Z. Ke and V. Ng, "Automated essay scoring: A survey of the state of the art," *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2019-Augus, pp. 6300–6308, 2019.

[10] A. J. Wohlpart, C. Lindsey, and C. Rademacher, "The Reliability of Computer Software to Score Essays: Innovations in a Humanities Course," *Computers and Composition*, vol. 25, no. 2, pp. 203–223, jan 2008.

[11] D. M. Williamson, X. Xi, and F. J. Breyer, "A Framework for Evaluation and Use of Automated Scoring," *Educational Measurement: Issues and Practice*, vol. 31, no. 1, pp. 2–13, 2012.

[12] E. Bosma, "Summary scoring using automatic machine translation evaluation techniques," Master's thesis, Utrecht University, 2021.

[13] F. Zoetmulder, "Automatic summary scoring using artificial neural networks," Master's thesis, Utrecht University, 2021.

[14] S. Vajjala, “Automated Assessment of Non-Native Learner Essays: Investigating the Role of Linguistic Features,” *Int J Artif Intell Educ*, vol. 28, pp. 79–105, 2018.

[15] E. Hovy, C. Y. Lin, L. Zhou, and J. Fukumoto, “Automated summarization evaluation with basic elements,” *Proceedings of the 5th International Conference on Language Resources and Evaluation, LREC 2006*, pp. 899–902, 2006.

[16] Wikipedia, “N-gram — Wikipedia, the free encyclopedia,” 2021, [Online; accessed 16-June-2021].

[17] F. Nicolás-Conesa, J. Roca de Larios, and Y. Coyle, “Development of EFL students' mental models of writing and their effects on performance,” *Journal of Second Language Writing*, vol. 24, no. 1, pp. 1–19, jun 2014.

[18] S. A. Crossley, K. Kyle, and D. S. Mcnamara, “Volume 8, Issue 1: 2015 To Aggregate or Not? Linguistic Features in Automatic Essay Scoring and Feedback Systems,” Tech. Rep. [Online]. Available: http://www.journalofwritingassessment.org/article.php?article=80

[19] NRC. [Online]. Available: https://www.nrc.nl

[20] I. Ben-Gal, “Outlier Detection,” in *Data Mining and Knowledge Discovery Handbook*. Springer-Verlag, may 2006, pp. 131–146. [Online]. Available: https://link-springer-com.proxy.library.uu.nl/chapter/10.1007/0-387-25465-X_7

[21] H. P. Vinutha, B. Poornima, and B. M. Sagar, “Detection of outliers using interquartile range technique from intrusion dataset,” in *Advances in Intelligent Systems and Computing*, vol. 701. Springer Verlag, 2018, pp. 511–518. [Online]. Available: https://doi.org/10.1007/978-981-10-7563-6_53

[22] N. Indurkhya and F. J. Damerau, *Handbook of natural language processing*. CRC Press, 2010, vol. 2.

[23] M. Honnibal and I. Montani, “spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing,” 2017, to appear.

[24] Stichting Opentaal, “Opentaal hunspell,” 2020. [Online]. Available: https://github.com/OpenTaal/opentaal-hunspell

[25] X. LU, “The Relationship of Lexical Richness to the Quality of ESL Learners' Oral Narratives,” *The Modern Language Journal*, vol. 96, no. 2, pp. 190–208, 2012.

[26] A. van Cranenburgh. Readability - pypi. [Online]. Available: https://https://pypi.org/project/readability/

[27] S. Štajner, R. Evans, C. Orasan, and R. Mitkov, "What can readability measures really tell us about text complexity," in *Proceedings of workshop on natural language processing for improving textual accessibility.* Citeseer, 2012, pp. 14–22.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] J. Friedman, T. Hastie, R. Tibshirani *et al.*, *The elements of statistical learning.* Springer series in statistics New York, 2001, vol. 1, no. 10.

[30] X. Lu, "Automatic analysis of syntactic complexity in second language writing," *International Journal of Corpus Linguistics*, vol. 15, pp. 474–496, 2010.

A Appendix

A.1 Python packages

Python version 3.7.10 was used for this research. All the used packages and the required versions can be found in table 4.

Package	Version
cyhunspell	2.0.2
import-ipynb	0.1.3
matplotlib	3.4.2
nl-core-news-lg	3.0.0
nltk	3.6.2
numpy	1.19.5
pandas	1.2.4
readability	0.3.1
scikit-learn	0.24.2
seaborn	0.11.1
spacy	3.0.6
wordcloud	1.8.1
yellowbrick	1.3.post1

Table 4: Python packages