SCRIPT: THESIS DEFENSE

1. **Introduction**

Hello everyone and welcome to my thesis defense. I am so glad to have you here.

Today I will be giving a small presentation of my thesis and would be happy to hear your feedback. In short, this work can be described as a tool that helps teachers grade essays more efficiently and provides instant feedback to students via visualisations.

1. **Motivation**

The motivation for this project is very straight-forward. Grading essays is very demanding and time consuming, which not only leads to high levels of stress for the teachers but also the feedback the students receive is often delayed and therefore not really helpful. With this thesis I mainly aim at help solving those two problems. But before I show you what I have done, let me briefly introduce you to what Automated Essay Scorers are.

1. **Background on Automated Essay Scorers**

AES are programs designed to take an essay as input and instantly output the grade most appropriate for the essay based on machine learning algorithms and natural language processing. However there has been a lot of outcry against them and their use for high stakes assessment. The two main points against AES are:

* Firstly, they are not reliable enough:

They do not take into account factual information, structure of the text, logical thinking and creativity among other things. Instead they base their predictions mostly on superficial linguistic features such as grammatical and stylistic errors while at the same time massively misidentify such mistakes

* Secondly, they tend to promote unauthentic writing:

When students know that no other human is going to read their work, students focus on just fooling the machine and getting a high score, which is a really easy to do. And overall the assessment loses its purpose.

1. **AES as a training Tool for Students**

Ok so AES are not exactly ready to replace human teachers, but can they be used as a training tool to help students improve their writing?

A study from last year had chinese students who were learning english as a foreign language use an online learning environment called pigai, that included an AES which students can use to see how the machine will grade their writing and what feedback they will receive.

The students reported that using the platformed really helped them improve, but only due to certain features. The highest rated feature was the thesaurus, which helped them learn new advanced words thus improving their vocabulary and help them avoid repetition. Identifying grammatical and spelling errors also contributed to students learning core language elements and avoid making mistakes in the future.

But the feedback was not perfect. Some feedback was given simply as: “suspected Chinglish”. Furthermore creativity and deviation from the norm was heavily punished. One example is an essay on the topic: "What would you buy if you had all the money in the world for one day and can not keep anything bought with those money after the end of the day." Almost everyone wrote something among the lines of driving expensive cars or going on luxurious holidays in a foreign land. One student however wrote that he would use the money to help collect all the garbage in the world and send it into space.

So what can we learn from all of this? That even for training it is best to keep a human reader. And that the automated feedback should be carefully selected and made easily digestible to students.

1. **Main Idea of this Thesis**

So now we arrive at the main idea of this thesis and that is to help create a tool that does not exclude a human reader, but rather tries to support him in the evaluation of essays and helps provide fast and reliable feedback to students at the same time. The novelty of the approach is that it does not simply use an AES to predict the score of an essay, but it uses it to help the human grader make more precise and time efficient decisions by presenting him with what the machine deems as most important. Furthermore additional features have been implemented, whose aim is to both further help our grader in his decision making and provide instant feedback to students to help them improve their writing based on the research gathered.

1. **Debugging an AES**

In this thesis the process of determining what the computer deems most important will be referred to as "Debugging a classifier",because grading an essay can be viewed as a classification problem.

But before we go any further I should talk a bit about Interpretability. Classification based on machine learning (which most of it is) takes into account so many different features in a way that humans are not traditionally used to (in the form of vectors, matrices, binary strings, etc.). That is why the explanations (the features the computer deems most important) will be presented as words and n-grams, so that everyone can get an understanding of the output without any previous knowledge.

Now that we have cleared this up the debugging is done based on the LIME algorithm developed at the Cornell University in 2016 (Local Interpretable Model-agnostic Explanations). The algorithm can be explained very simply:

1.Generating distorted version of the text, that we want to grade

2.Pass those texts into the classifier

3.Train another classifier based on a more simple linear regression model with the input and output of our distorted texts

4.Get the weight of the feature that we want

The algorithm works because although it might be difficult to understand how a classifier is trained based on a large dataset. It is fairly easy to see how it reacts to a small sample based on a given text.

1. **Additional Features**

Although the thesis is build around this approach and this is the main focus, there are also some additional features added to give a better feeling of what the complete product should look like and offer.

1. **Building a classifier**

For this thesis I really wanted to use one of the commercially available ES (Project Essay Grade/ e-rater/ Intelligent Essay Assessor), but since none of them is neither open source nor free I could not. That is why for the purpose of this thesis I have build my own essay scorer. It is based on a natural language processing technique called Latent Semantic Analysis (LSA) that is also used in one of the aforementioned softwares and aims at exploring the relationship between a set of documents and the words they contain.

1. **Latent Semantic Analysis(LSA)**

LSA works by reducing documents to vectors where every term in the document is represented by a weight. A term can be either a word or an n-gram. The weight is determined based on the term’s frequency inside the document and the inverse document frequency. This is done to reduce the weight of common stop words, and to make sure that rarer terms are upweighted.

After all the documents have been turned into vectors, they are being reduced by a method called Truncated Singular Value Decomposition (TSVD), which leaves only the k-most expressed features.

After the texts has been processed to create our essay scorer I have used a machine learning method called Support Vector Machine (SVM).

1. **Support Vector Machine**

SVM is a great machine learning method for data that can be represented as high dimensional vectors. If we view each text as a point in a k dimensional space the SVM algorithm tries to find a k-1 dimensional hyperplane that separates our five “classes” with the largest margin possible. Because SVM is traditionally a binary classification method, to use it for our purpose I have used a one vs one approach, where for each unique pair of possible grades one is selected over the other and its probability score is increased. At the end, the grade with the higher score is chosen as most likely.

Now that we have all the necessary preparations for our essay scorer model we just need to feed it data so that it can learn.

1. **Dataset Used**

The data used to train and test this classifier and the overall product is the ASAP (Automated Student Assessment Prize) Dataset, presented by The Hewlett Foundation. It consists of 8 different essay sets, each set based on a different assignment. Some of them are based on a given source and others give students the freedom to write about their own personal experiences. All information that can be considered personally identifying has been anonymised to protect the identities of the students. Furthermore each set has a different scoring range, so I have normalised all of them to fit the standard 5 grade scale, where 1 represents the worst possible grade and 5 the best.

1. **Evaluation of the Classifier**

Before I give you a preview of the application, there is one last thing that needs to be discussed. That is the performance of the classifier that I previously mentioned. Although this is not the main purpose of our essay scorer and this thesis, we still need to see how goods its predictions are, before we try to explain them. I have generated heat-map confusion matrices for each data set, where the correct predictions are along the diagonal and the others are mismatches. Furthermore I have generated additional metrics such as accuracy, precision, recall and f1-score.

Accuracy = # correct predictions / # predictions in total

Precision = #true positives reported/ # total positives reported (how many predicted items are relevant) Precision is the average probability of relevant retrieval

Recall = #true positives reported / # true positives overall (how many relevant items are selected) Recall is the estimated probability that a randomly selected relevant document is retrieved in a search.

F-score = 2x(PxR/P+R) (the balance between P and R: harmonic mean)

Precision shows us how many selected items are correctly belonging to that class, but that does not mean that all items belonging to class X are selected.

Also recall shows us how many items of class X are selected correctly, but not how many items are incorrectly marked as belonging to class X.

1. **Quick Preview**

Now let me just give you a quick preview of the application. It is very easy and simple to use. First we need to upload a data set to train our essay scorer. After this we simply select the essay that we want to grade and wait to see the dashboard. Another advantage to this application that I have build is the flexibility is that a teacher can hand select the training data set and modify it as he pleases. Whereas when using one of the software that are available on the market, this is not possible.

The training set should be in a tab separated value file format. This is simply because the data was published this way, but it can easily be extended to work with other table like formats, such as excel file or comma separated ones.

The essay are submitted as plain text files for the same reason. But this also can be made to accept PDF, Microsoft word files, and other popular formats.

1. **Feedback**

And before I give you the word I just want to quickly mention some feedback that I received from colleagues of mine that tested the application and share my thoughts on what future direction such a project can take. Firstly the essay scorer model can be improved to predict grades with higher precision, furthermore additional features can be added such as a thesaurus and a detection for grammar mistakes such as in grammarly. Furthermore the highlights of the text can be based on more objective criteria, such as the English Profile Project.

And finally that such a tool should be implemented into an Online Learning Environment, so that students can actually receive their automated feedback instantly after submitting.

Thank you very much for your attention!