Essay Automatic Grading

Introduction:

Manual Assessment of student's essay writing and providing thoughtful feedback is a truly labour-intensive and time-consuming task. With human instructors already overwhelmed, the alternate is to consider a computer-based grading. The purpose to develop this NLP application is to facilitate the institutes to summarize and grade essay or short answers automatically written by students. The proposed model will provide the best possible result and performance. Study shows that AI based automatic text summarization is more accurate than the summarization done by human. AI based summarization and grading system minimize or almost remove human involvement.

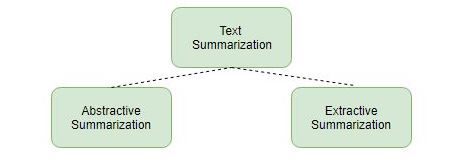
Theory behind NLP and text summarization

***“Automatic text summarization is the task of producing a concise and fluent summary while preserving key information content and overall meaning”***

***-Text Summarization Techniques: A Brief Survey, 2017***

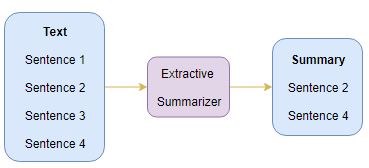
There are broadly two different approaches that are used for text summarization:

* Extractive Summarization
* Abstractive Summarization



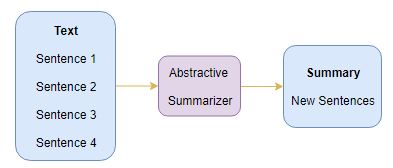
**Extractive Summarization**

The name gives away what this approach does. **We identify the important sentences or phrases from the original text and extract only those from the text.** Those extracted sentences would be our summary. The below diagram illustrates extractive summarization:



### Abstractive Summarization

This is a very interesting approach. Here, we generate new sentences from the original text. This is in contrast to the extractive approach we saw earlier where we used only the sentences that were present. The sentences generated through abstractive summarization might not be present in the original text:



**Detail steps in NLP**

**Lexical Analysis-** collect the words

**Syntactical Analysis** - use words to form sentences. Irrespective of its meaning.

**Semantic Analysis** - try to understand the meaning of sentences

**Disclosure integration**- Checks the references between different sentences.

**Pragmatic Analysis -** Overall supervising to get most appropriate result.

**Techniques converting sentence into vectors**

1. **Count vectorizer**
2. **TF-IDF**
3. **Hashing Vectorizer**
4. **Bag of words**
5. **Word2vec - Skipgram, CBOW**

The word2vec algorithm uses a [neural network](https://en.wikipedia.org/wiki/Neural_network) model to learn word associations from a large corpus of text.

1. Skip-gram,

**2. CBOW**

1. **GloVe (Global Vector)** The model is an [unsupervised learning](https://en.wikipedia.org/wiki/Unsupervised_learning) algorithm for obtaining vector representations for words.
2. **Bert:** Transformer based pre-trained model

**Vector comparison techniques**

1. **cosine similarity,**
2. **Euclidean distance,**
3. **Jaccard distance,**
4. **word mover’s distance**

**Data Set**

Training Dataset:

There are 10 datasets. Each dataset represents an Essay set. Students primarily in Grade 10 wrote all responses. All responses were hand graded and were double-scored.

Training data is provided in tsv form (Tab Separated Value)

1. File Name: train.tsv  
   Size: 17207 X 5
2. File Name: train\_rel\_2.tsv (Some of the records are removed because of  
   transcription error)  
   Size: 17043 X 5

**Features:**

1. Id: A unique identifier for each individual student essay.  
2. EssaySet: 1-10, an id for each set of essays.  
3. Score1: The human rater's score for the answer. This is the final score for the answer  
and the score that you are trying to predict.  
4. Score2: A second human rater's score for the answer. This is provided as a measure  
of reliability, but had no bearing on the score the essay received.  
5. EssayText: The ascii text of a student's response

Test Dataset:

I. File Name: test1.tsv  
Size: 5224 X 3

**Features:**

1. Id:  
2. EssaySet:  
3. EssayText:

II. File Name: test1\_soln.csv (Dataset having SCORE for the test data test1.csv)  
Size: 5224 X 4

**Features:**

1. Id:  
2. EssaySet:  
3. EssayWeight:  
4. EssayScore:

III. File Name: test2.tsv (Dataset without having SCORE)  
Size: 5100 X 3

**Features:**

1. Id:  
2. EssaySet:  
3. EssayText:

Technical specifications:

flask

Keras-Preprocessing

nltk

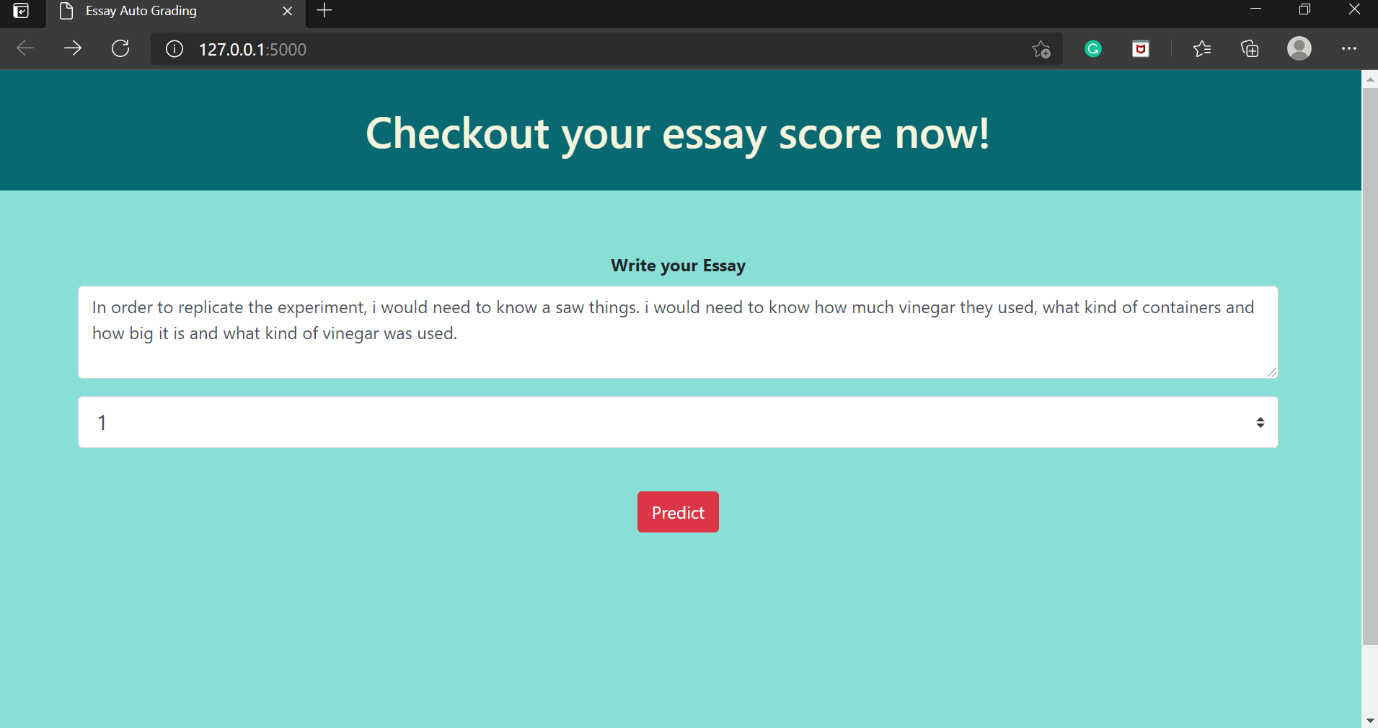
pickle5

pymongo

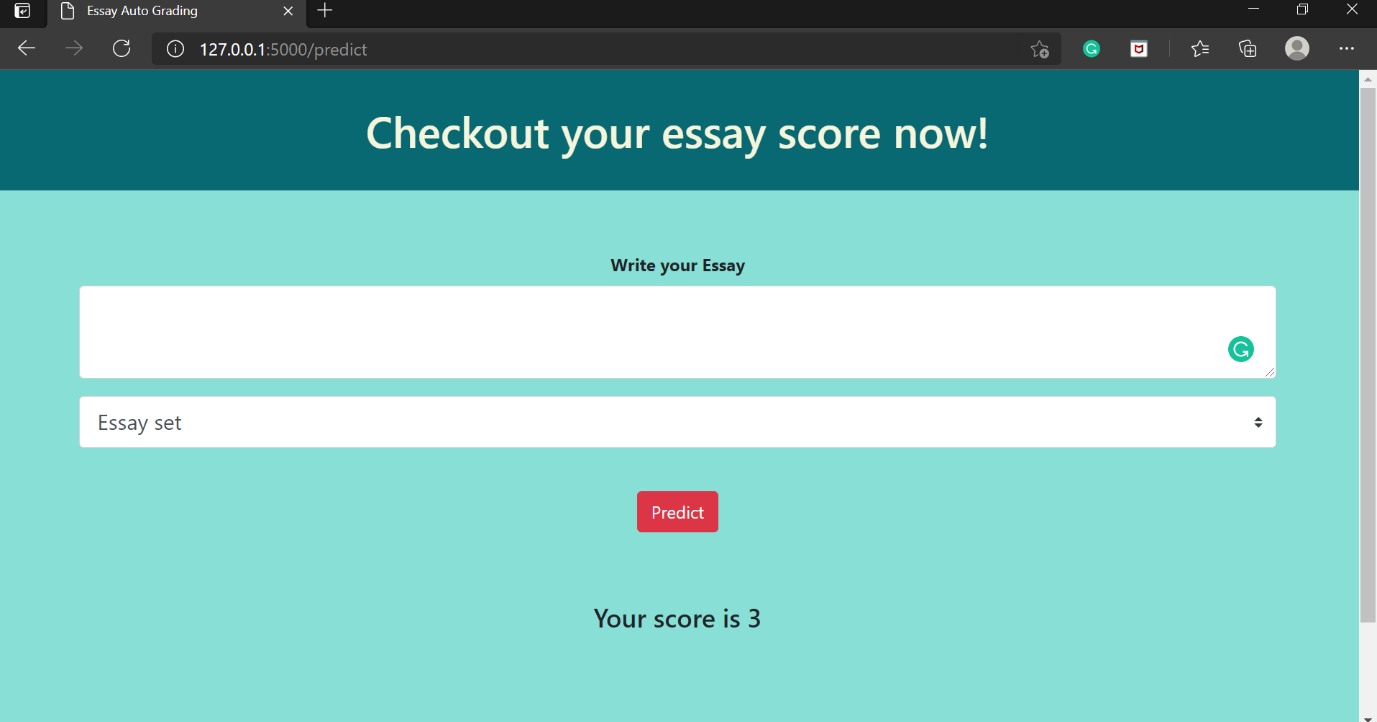
tensorflow

Python

Application Interface



Home.html



Result.html

Related work

1. A system to automatically grade the school children essays in Arabic, calling it AAEE for “automatic Arabic essays evaluator” was proposed.. The system was modeled upon the scoring scheme followed by the school instructors in Saudi Arabia. The instructors had specific criteria upon which an essay is assessed. Putting these criteria together, they developed a system that relies on Latent Semantic Analysis, and Rhetorical Structure Theory.
2. Burstein & Marcu (2000) argued that vocabulary in 40% and 60% discourse-based summaries contain considerable information from the original essay. They hypothesize that essays are written under time constraint, leaving little time for the student to edit and proof read the text. As a result, essays could have a lot of noise, such as repetitive statements and extraneous comments irrelevant to the main arguments in the essay. Summaries, the author’s state, are able to clean up the essay by removing some of the noise leaving the more important part out. This technique could be used to provide feedback for subject-based essays such as in College Board’s AP exams in US History.
3. In (Bin et al., 2008) proposed applying k-Nearest Neighbor (k-NN) used in text categorization model to essay scoring. Implementing the k-NN algorithm for AEE systems occurs in steps. First step, transforming the problem into text categorization involves transforming the required scoring to categories. Next, transforming the essays into vectors. This is followed by feature selection and reduction to minimize the vector space dimensions. The authors used two methods for feature selection: term frequency (TF) and information gain (IG). Search for the k-NN of the training essays and computing the nearest objects from the new essay to all essays in the training set. After that, the results are sorted in descending order, and the first k essays are selected. The final step is text categorization, which involves classifying the essay into the same category as that of nearby essays. Experiments on CET4 essays in the Chinese Learner English Corpus show achieving accuracy above 76%. In this study, the best result is achieved when k = 3.
4. In (Kakkonen et al., 2008) looked into three different dimension reduction schemes, namely: latent semantic analysis (LSA), probabilistic LSA (PLSA), and latent Dirichlet allocation (LDA); on automated essay grading. All three schemes assume that we can model documents as mere collection of words without any regard to their order. Following experiments, the authors concluded that LSA yield slightly more accurate essay grades than the other two schemes. However, PLSA and LDA tend to provide better means in giving students feedback on the essay.

References

1. Azmi, Al-Jouie, and Hussain, Information Processing & Management (2019), vol. 56, no. 5, pp. 1736-1752. (<https://www.researchgate.net/publication/334401130_AAEE-Automated_evaluation_of_students'_essays_in_Arabic_language>)
2. Bin, L., Jun, L., Jian-min, Y., & Qiao-ming, Z. (2008). Automated essay scoring using the KNN algorithm. In International Conference on Computer Science and Software Engineering (pp. 735–738). ()
3. <https://www.analyticsvidhya.com/blog/2018/11/introduction-text-summarization-textrank-python/>
4. https://www.aclweb.org/anthology/C18-1094.pdf