Automated Essay Grading System

Team Name : Vectors
Group Members: Mohit Sharma(11840730)
Vivek Singh(11841240)

Abstract :

Essays are paramount for assessing academic excellence along with linking the different ideas with the ability to recall but are notably time consuming when they are assessed manually. Manual grading takes a significant amount of evaluator's time and hence it is an expensive process. Automated grading if proven effective will not only reduce the time for assessment but comparing it with human scores will also make the score realistic. The purpose of this project is to implement and train machine learning models to automatically assess and grade essay responses. The dataset we used is made available on the internet by Hewlett Foundation for one competition on Kaggle.com. We used a linear regression model for training our model. The features extracted from essays include number of sentences, words, comma, exclamation mark, brackets, nouns, verbs, adjectives and misspelled words etc. We tested our model on a testing dataset and calculated the RMSE(Root Mean Square Error) for each essay set. We have found that RMSE values for set 1-6 are lower than set 7 and 8, this is because essays in these sets are much more content specific. **In the improvement/ optimization phase of the project, we added more advanced NLP features to our project namely Bag of Words, Bigram and Trigram.**

Improvements from the previous version of the project :

In the improvement/ optimization phase of the project, we added more advanced NLP features to our project namely Bag of Words, Bigram and Trigram. We have also added lexical diversity in this phase of the project and applied cross validation. Because of the addition of more advanced NLP features along with lexical diversity etc. the overall performance of the system has improved and now its working better even for content specific essays also.

Bag of Words:

The bag-of-words model is a way of representing text data when modeling text with machine learning algorithms. A bag-of-words is a representation of text that describes the occurrence of words within a document.

It involves two things:

- A vocabulary of known words.
- A measure of the presence of known words.

It is called a “bag” of words, because any information about the order or structure of words in the document is discarded. The intuition is that documents are similar if they have similar content.

Example
Below is a snippet of the first few lines of text:

It was the best of times,
it was the worst of times,
it was the age of wisdom,

it was the age of foolishness,

Now we can make a list of all of the words in our model vocabulary. The unique words here (ignoring case and punctuation) are:

- "it"
- "was"
- "the"
- "best"
- "of"
- "times"
- "worst"
- "age"
- "wisdom"
- "foolishness"

That is a vocabulary of 10 words from a corpus containing 24 words.

The scoring of the first line of the document would look as follows:

- "it" = 1
- "was" = 1
- "the" = 1
- "best" = 1
- "of" = 1
- "times" = 1
- "worst" = 0
- "age" = 0
- "wisdom" = 0
- "foolishness" = 0

```
feature = feature.fillna(0)
feature.iloc[0:10,30:50]
```

	famili	friend	game	good	great	help	inform	internet	just	kid	know	learn	life	like	live	look	lot	make
0	0.0	2.0	1.0	0.0	2.0	2.0	0.0	1.0	2.0	0.0	2.0	6.0	0.0	0.0	0.0	0.0	1.0	0
1	0.0	4.0	0.0	0.0	1.0	8.0	3.0	2.0	0.0	0.0	2.0	0.0	2.0	6.0	0.0	0.0	2.0	3
2	1.0	0.0	0.0	0.0	2.0	2.0	0.0	0.0	0.0	1.0	4.0	0.0	0.0	0.0	2.0	0.0	2.0	1
3	0.0	0.0	0.0	5.0	0.0	7.0	1.0	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
4	11.0	0.0	2.0	1.0	1.0	4.0	3.0	2.0	5.0	0.0	2.0	3.0	0.0	0.0	1.0	0.0	0.0	1
5	0.0	1.0	0.0	2.0	0.0	0.0	2.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0
6	3.0	3.0	1.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	3.0	0.0	4.0	1.0	0.0	0.0	0.0	0
7	1.0	1.0	0.0	2.0	0.0	1.0	4.0	0.0	1.0	0.0	3.0	3.0	2.0	1.0	0.0	1.0	6.0	4
8	1.0	3.0	4.0	1.0	1.0	0.0	0.0	1.0	4.0	1.0	1.0	0.0	1.0	1.0	2.0	2.0	0.0	2
9	1.0	1.0	0.0	2.0	1.0	0.0	0.0	0.0	2.0	8.0	0.0	0.0	1.0	2.0	3.0	0.0	1.0	2

Bag of words of essay set

Bi-gram and tri-gram:

Bi-gram and tri-gram are similar to bag of words. Here vocabulary is created by taking ordered pair of words in the document. For example some bi-grams from above document are 'it was', 'the best', 'the age' etc. Some tri-grams are 'best of times', 'worst of times', 'age of wisdom' etc. Scoring of document is done in similar manner.

```
feature = feature.fillna(0)
feature.iloc[6060:6070,170:200]
```

	end stori	fail	flower	gee	gee return	gee return hibiscus	hibiscus	hibiscus bud	hibiscus bud test	home	melt	melt gee	melt gee return	mother
6060	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0
6061	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
6062	1.0	1.0	1.0	2.0	0.0	0.0	1.0	0.0	0.0	3.0	0.0	0.0	0.0	0.0
6063	1.0	1.0	0.0	1.0	1.0	0.0	2.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6064	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6065	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
6066	1.0	0.0	0.0	1.0	0.0	0.0	4.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
6067	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6068	1.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0
6069	1.0	0.0	0.0	2.0	1.0	0.0	1.0	1.0	0.0	3.0	1.0	1.0	1.0	0.0

Bi-gram and Tri-gram of essay set

Lexical Diversity:

Lexical diversity is one aspect of 'lexical richness' and refers to the ratio of different unique word stems (types) to the total number of words (tokens).

Stemming and lemmatization:

The goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form. For instance:

am, are, is =>be
car, cars, car's, cars' $\Rightarrow$ car

The result of this mapping of text will be something like:

the boy's cars are different colors $\Rightarrow$ the boy car be differ color

For grammatical reasons, documents are going to use different forms of a word, such as *organize*, *organizes*, and *organizing*. Additionally, there are families of derivationally related words with similar meanings, such as *democracy*, *democratic*, and *democratization*. In many situations, it seems as if it would be useful for a search for one of these words to return documents that contain another word in the set.

```
from nltk.stem import SnowballStemmer
stemmer = SnowballStemmer("english",ignore_stopwords=True)


w_tokenizer = nltk.tokenize.WhitespaceTokenizer()

def stemming_text(text):
    return [" ".join(stemmer.stem(w) for w in w_tokenizer.tokenize(text))]

df_train_2 = df_train_2.apply(stemming_text)
```

Final Results:

After adding bag of words, bi-gram, tri-gram and lexical diversity as features there is improvement in set 7 and set 8 essays. RMSE value of set 7 and set 8 without extra features was 3.29 and 4.27 respectively. RMSE after adding extra features is 3.1 and 3.56. Thus we can conclude that the overall performance of the system is improved as compared to its previous version.

```
RMSE[:][:]
```

	Essay_set	Marking_Scale	RMSE	Max_AE	Agreement with RMSE
0	1	2-12	0.906868	3.89717	71.641791
1	2	1-5	0.567178	1.95887	69.144981
2	3	0-3	0.569921	1.94656	74.031008
3	4	0-3	0.645312	2.24472	72.075472
4	5	0-4	0.509169	1.53459	66.666667
5	6	0-4	0.574164	1.8262	67.286245
6	7	0-30	3.10153	9.23852	63.829787
7	8	0-60	3.56444	9.77125	69.444444