

Intelligent-test-interface-with-plagiarism-detection

Generated by Doxygen 1.8.13

Contents

1	Namespace Index	1
1.1	Namespace List	1
2	File Index	3
2.1	File List	3
3	Namespace Documentation	5
3.1	generate_csv Namespace Reference	5
3.1.1	Function Documentation	5
3.1.1.1	generate_csv()	5
3.1.2	Variable Documentation	6
3.1.2.1	args	6
3.1.2.2	help	6
3.1.2.3	parser	6
3.2	Match_strings Namespace Reference	6
3.2.1	Function Documentation	7
3.2.1.1	cosangle()	7
3.2.1.2	dotproduct()	7
3.2.1.3	length()	7
3.2.1.4	met_cosine()	8
3.2.1.5	met_cosine_word()	8
3.2.1.6	met_jaccard()	9
3.2.1.7	met_lcs()	9
3.2.1.8	plag_index_strings()	9

3.2.1.9	<code>pre_process()</code>	10
3.3	plagiarism Namespace Reference	10
3.3.1	Function Documentation	11
3.3.1.1	<code>extract_answers()</code>	11
3.3.1.2	<code>extract_questions()</code>	11
3.3.1.3	<code>extract_roll()</code>	12
3.3.1.4	<code>mult_corr_list()</code>	12
3.3.1.5	<code>norm_final_index()</code>	13
3.3.1.6	<code>plag_calc()</code>	13
3.3.1.7	<code>plag_ques_wise()</code>	14
3.3.2	Variable Documentation	14
3.3.2.1	<code>cursor</code>	14
3.3.2.2	<code>db</code>	14
4	File Documentation	15
4.1	<code>generate_csv.py</code> File Reference	15
4.1.1	Detailed Description	15
4.2	<code>Match_strings.py</code> File Reference	16
4.2.1	Detailed Description	16
4.3	<code>plagiarism.py</code> File Reference	17
4.3.1	Detailed Description	17

Chapter 1

Namespace Index

1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

generate_csv	5
Match_strings	6
plagiarism	10

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

generate_csv.py	This file of the project takes some required files from user and converts those files to array or list formats so as to take in use afterwards	15
Match_strings.py	This file of the project is applying various string matching algorithms for detecting plagiarism between subjective type questions	16
plagiarism.py	This file of the project extracts stored data of responses from the database and evaluates a normalised plagiarism index	17

Chapter 3

Namespace Documentation

3.1 generate_csv Namespace Reference

Functions

- def `generate_csv` (parentID, solutions, weights)
This function takes testID, solutions csv path and weights csv path as an arguments and generating the output plagiarism index in a #csv file.

Variables

- `parser` = argparse.ArgumentParser(description = 'Say hello')
- `help`
- `args` = parser.parse_args()

3.1.1 Function Documentation

3.1.1.1 generate_csv()

```
def generate_csv.generate_csv (
    parentID,
    solutions,
    weights )
```

This function takes testID, solutions csv path and weights csv path as an arguments and generating the output plagiarism index in a #csv file.

In this the solutions csv file is first structured in a format so the other functions can compare the answers given by students with correct answer efficiently. Also the weights are being extracted from its location and normalized in this function. Finally all these are passed to another function that is checking plagiarism and returning a numpy array having the plagiarism index.

Parameters

<i>parentID</i>	int
<i>solutions</i>	str
<i>weights</i>	str

3.1.2 Variable Documentation

3.1.2.1 args

```
generate_csv.args = parser.parse_args()
```

3.1.2.2 help

```
generate_csv.help
```

3.1.2.3 parser

```
generate_csv.parser = argparse.ArgumentParser(description = 'Say hello')
```

3.2 Match_strings Namespace Reference

Functions

- def [pre_process](#) (s)
This function is taking a String as an argument, truncating spaces and converting it to a lowercase substring as a whole.
- def [dotproduct](#) (v1, v2)
This function is taking two vectors as arguments and returning a single dot product value for them.
- def [length](#) (v)
This function is taking a vector as argument and returning a single value which is norm of it.
- def [cosangle](#) (v1, v2)
This function is taking two vectors as arguments and returning a single cosine value of the angle between these vectors.
- def [met_cosine_word](#) (s1, s2, n)
This function is taking two vectors as arguments and returning a single cosine value of the angle between these vectors.
- def [met_jaccard](#) (s1, s2, n)
This function is taking two strings as arguments and returning a single value which is jaccard index between these two strings.
- def [met_cosine](#) (s1, s2, n)
This function is taking two strings as arguments and returning a single value which is cosine metric index between these two strings.
- def [met_lcs](#) (s1, s2)
This function is taking two strings as arguments and returning metric similarity between them.
- def [plag_index_strings](#) (str1, str2, weight)
This function is taking two strings as arguments and one list consisting of weights int it corresponding to each algorithm.

3.2.1 Function Documentation

3.2.1.1 cosangle()

```
def Match_strings.cosangle (
    v1,
    v2 )
```

This function is taking two vectors as arguments and returning a single cosine value of the angle between these vectors.

Returns

Cosine of the angle between the two vectors

Parameters

V1	vector 1
V2	vector 2

3.2.1.2 dotproduct()

```
def Match_strings.dotproduct (
    v1,
    v2 )
```

This function is taking two vectors as arguments and returning a single dot product value for them.

Returns

Dot product of both the arguments

Parameters

V1	vector 1
V2	vector 2

3.2.1.3 length()

```
def Match_strings.length (
    v )
```

This function is taking a vector as argument and returning a single value which is norm of it.

Returns

Norm of the vector

Parameters

<i>V1</i>	vector 1
-----------	----------

3.2.1.4 met_cosine()

```
def Match_strings.met_cosine (
    s1,
    s2,
    n )
```

This function is taking two strings as arguments and returning a single value which is cosine metric index between these two strings.

Returns

cosine metric index between given strings

Parameters

<i>s1</i>	String
<i>s2</i>	String
<i>n</i>	int met_cosine will take n letters at a time

3.2.1.5 met_cosine_word()

```
def Match_strings.met_cosine_word (
    s1,
    s2,
    n )
```

This function is taking two vectors as arguments and returning a single cosine value of the angle between these vectors.

Returns

Cosine of the angle between two vectors formed by taking frequency of words instead of characters

Parameters

<i>s1</i>	String
<i>s2</i>	String
<i>n</i>	int cosine angle will take n letters at a time

3.2.1.6 met_jaccard()

```
def Match_strings.met_jaccard (
    s1,
    s2,
    n )
```

This function is taking two strings as arguments and returning a single value which is jacard index between these two strings.

Returns

Jacard index between given strings

Parameters

<i>s1</i>	String
<i>s2</i>	String
<i>n</i>	int jacard index will take n letters at a time

3.2.1.7 met_lcs()

```
def Match_strings.met_lcs (
    s1,
    s2 )
```

This function is taking two strings as arguments and returning metric similarity between them.

Returns

Metric similarity between two strings

Parameters

<i>s1</i>	String
<i>s2</i>	String

3.2.1.8 plag_index_strings()

```
def Match_strings.plag_index_strings (
    str1,
```

```

    str2,
    weight )

```

This function is taking two strings as arguments and one list consisting of weights int it corresponding to each algorithm.

It evaluates string matching according to given weights of algorithms and finally return us with an index.

Returns

ans float plagiarism index between two responses submitted by 2 students

Parameters

<i>s1</i>	String
<i>s2</i>	String
<i>weight</i>	list it is the list set by test taker to restrict what weihtage should be given to each algorithm.

3.2.1.9 pre_process()

```

def Match_strings.pre_process (
    s )

```

This function is taking a String as an argument, truncating spaces and converting it to a lowercase substring as a whole.

Returns

String

Parameters

<i>s</i>	String
----------	--------

3.3 plagiarism Namespace Reference

Functions

- def [plag_calc](#) (parent_id, correct, marks, weight)
This function is taking a filename as argument and returning a list of integers inside the file.
- def [extract_roll](#) (test_id)
This function is taking a test id as argument and returning a list of roll numbers attempted that test.
- def [extract_questions](#) (test_id)
This function takes test id as argument and returns a list of list which consist of triplets id, label and type of each question.

- def `extract_answers` (roll, ques)
This function takes list of roll numbers as argument and returns a list of list which contains all answers.
- def `plag_ques_wise` (roll, ques, ans, weight, correct)
Taking rollnumbers, questions, answers, weights and correct sols and returning the plagarium index.
- def `norm_final_index` (index_q, marks)
This function takes plagiarism indices for individual questions and weight them according to the distribution of marks.
- def `mult_corr_list` (ans)
This function takes answers stored for multiple choice MCQs and format that into sorted order of options.

Variables

- `db` = pymysql.connect("localhost","wordpressuser","wordpressuser","wordpress")
- `cursor` = db.cursor()

3.3.1 Function Documentation

3.3.1.1 `extract_answers()`

```
def plagiarism.extract_answers (
    roll,
    ques )
```

This function takes list of roll numbers as argument and returns a list of list which contains all answers.

For each roll number answer is having a list in it which consist of this roll number's responses to all the questions of the test. Iterating the same way answer is list of lists which keeps the information of responses of all the roll numbers attempted this test

Returns

answers list of lists contains all question's responses corresponding to all the roll numbers

Parameters

<i>roll</i>	list roll number
<i>ques</i>	list of lists information about the question extracted from database

3.3.1.2 `extract_questions()`

```
def plagiarism.extract_questions (
    test_id )
```

This function takes test id as argument and returns a list of list which consist of triplets id, label and type of each question.

Function is executing a query and extracting only questions (Submit label and roll number are ignored) from the database corresponding to that test. It is also storing the type of the question.

Returns

questions list of lists

Parameters

<i>test_id</i>	int test id
----------------	-------------

3.3.1.3 extract_roll()

```
def plagiarism.extract_roll (
    test_id )
```

This function is taking a test id as argument and returning a list of roll numbers attempted that test.

Since test id field is not there in the table wp_postmeta, this function is extracting roll numbers by first extracting id values corresponding to that test id and whose label is roll number or similar to it. Then using these id values roll numbers can be extracted from wp_postmeta table.

Returns

roll_numbers list

Parameters

<i>test_id</i>	int test id
----------------	-------------

3.3.1.4 mult_corr_list()

```
def plagiarism.mult_corr_list (
    ans )
```

This function takes answers stored for multiple choice MCQs and format that into sorted order of options.

Returns

res list consist of selected options in sorted order

Parameters

<i>ans</i>	String specifies choices selected by student
------------	--

3.3.1.5 norm_final_index()

```
def plagiarism.norm_final_index (
    index_q,
    marks )
```

This function takes plagiarism indices for individual questions and weight them according to the distribution of marks.

For each pair of roll numbers there is a plagiarism index for each question. To get one normalised Plagiarism index, these indices are weighted according to the Question's marks distribution and It finally return us a final normalised Plagiarism Indices for all pairs.

Returns

final list Contains normalised plagiarism index for each pair of roll numbers

Parameters

<i>index_x</i>	list Contains plagiarised index corresponding to each question for each pair of rollnumbers
<i>marks</i>	list marks weightage given by the teacher to each question

3.3.1.6 plag_calc()

```
def plagiarism.plag_calc (
    parent_id,
    correct,
    marks,
    weight )
```

This function is taking a filename as argument and returning a list of integers inside the file.

The given file contains integers each line. This function is picking out text line by line, typecasting it to integers and appending it to a list. Finally it is returning that list.

Returns

l list

Parameters

<i>filename</i>	the name of the file
-----------------	----------------------

3.3.1.7 plag_ques_wise()

```
def plagiarism.plag_ques_wise (
    roll,
    ques,
    ans,
    weight,
    correct )
```

Taking rollnumbers, questions, answers, weights and correct sols and returning the plagarium index.

This function takes all nC2 combinations of the students who has taken the test and comparing their solutions with each other. And then evaluating a cheating index for that question. Also there is a progress bar at the end which shows the % of job finished.

Returns

plag_index

Parameters

<i>roll</i>	list All rollnumbers
<i>ques</i>	list All questions
<i>ans</i>	list All answers
<i>weight</i>	list Weights of algorithms
<i>correct</i>	list

3.3.2 Variable Documentation

3.3.2.1 cursor

```
plagiarism.cursor = db.cursor()
```

3.3.2.2 db

```
plagiarism.db = pymysql.connect("localhost","wordpressuser","wordpressuser","wordpress" )
```

Chapter 4

File Documentation

4.1 generate_csv.py File Reference

This file of the project takes some required files from user and converts those files to array or list formats so as to take in use afterwards.

Namespaces

- [generate_csv](#)

Functions

- def [generate_csv.generate_csv](#) (parentID, solutions, weights)
This function takes testID, solutions csv path and weights csv path as an arguments and generating the output plagiarism index in a #csv file.

Variables

- [generate_csv.parser](#) = argparse.ArgumentParser(description = 'Say hello')
- [generate_csv.help](#)
- [generate_csv.args](#) = parser.parse_args()

4.1.1 Detailed Description

This file of the project takes some required files from user and converts those files to array or list formats so as to take in use afterwards.

This file is basically taking some arguments from the test taker to identify test uniquely, that test's results and then weights #teacher wants to give to different algorithms. These inputs are further processed to take in use afterwards.

Author

Infernos : CS699 Course Project Software Lab

Date

Thursday, November 29, 2019

4.2 Match_strings.py File Reference

This file of the project is applying various string matching algorithms for detecting plagiarism between subjective type questions.

Namespaces

- [Match_strings](#)

Functions

- def [Match_strings.pre_process](#) (s)
This function is taking a String as an argument, truncating spaces and converting it to a lowercase substring as a whole.
- def [Match_strings.dotproduct](#) (v1, v2)
This function is taking two vectors as arguments and returning a single dot product value for them.
- def [Match_strings.length](#) (v)
This function is taking a vector as argument and returning a single value which is norm of it.
- def [Match_strings.cosangle](#) (v1, v2)
This function is taking two vectors as arguments and returning a single cosine value of the angle between these vectors.
- def [Match_strings.met_cosine_word](#) (s1, s2, n)
This function is taking two vectors as arguments and returning a single cosine value of the angle between these vectors.
- def [Match_strings.met_jaccard](#) (s1, s2, n)
This function is taking two strings as arguments and returning a single value which is jaccard index between these two strings.
- def [Match_strings.met_cosine](#) (s1, s2, n)
This function is taking two strings as arguments and returning a single value which is cosine metric index between these two strings.
- def [Match_strings.met_lcs](#) (s1, s2)
This function is taking two strings as arguments and returning metric similarity between them.
- def [Match_strings.plag_index_strings](#) (str1, str2, weight)
This function is taking two strings as arguments and one list consisting of weights int it corresponding to each algorithm.

4.2.1 Detailed Description

This file of the project is applying various string matching algorithms for detecting plagiarism between subjective type questions.

In this analysis of various string matching algorithms implemented in strsim library is done and then the algorithms whose performance is good enough for the task are used combinely to come up with a fair match index between strings.

Author

Infernos : CS699 Course Project Software Lab

Date

Thursday, November 29, 2019

4.3 plagiarism.py File Reference

This file of the project extracts stored data of responses from the database and evaluates a normalised plagiarism index.

Namespaces

- [plagiarism](#)

Functions

- def [plagiarism.plag_calc](#) (parent_id, correct, marks, weight)
This function is taking a filename as argument and returning a list of integers inside the file.
- def [plagiarism.extract_roll](#) (test_id)
This function is taking a test id as argument and returning a list of roll numbers attempted that test.
- def [plagiarism.extract_questions](#) (test_id)
This function takes test id as argument and returns a list of list which consist of triplets id, label and type of each question.
- def [plagiarism.extract_answers](#) (roll, ques)
This function takes list of roll numbers as argument and returns a list of list which contains all answers.
- def [plagiarism.plag_qes_wise](#) (roll, ques, ans, weight, correct)
Taking rollnumbers, questions, answers, weights and correct sols and returning the plagiarism index.
- def [plagiarism.norm_final_index](#) (index_q, marks)
This function takes plagiarism indices for individual questions and weight them according to the distribution of marks.
- def [plagiarism.mult_corr_list](#) (ans)
This function takes answers stored for multiple choice MCQs and format that into sorted order of options.

Variables

- [plagiarism.db](#) = pymysql.connect("localhost","wordpressuser","wordpressuser","wordpress")
- [plagiarism.cursor](#) = db.cursor()

4.3.1 Detailed Description

This file of the project extracts stored data of responses from the database and evaluates a normalised plagiarism index.

This is the python script which will automatically check for plagiarism. For each roll number all it's answers to the test questions are extracted from the database. Then for each question's answer, his response is checked pairwise with all the other responses submitted for the same test. Plagiarism indices corresponding to each answer with each other student's response are stored in output csv. And finally weighted plagiarism index according to the distribution of marks of the questions is evaluated and returned.

Author

Infernos : CS699 Course Project Software Lab

Date

Thursday, November 29, 2019

