A New Similarity Measure with Length Factor for Plagiarism Detection

Dhrubajyoti Baruah,
Assistant Professor, Dept. of Computer Application,
Jorhat Engineering College,
Jorhat, Assam, India. PIN: 785007

Anjana Kakoti Mahanta,Ph.D
Professor, Computer Science Dept.,
Gauhati University,
Guwahati,Assam, India. PIN: 781014

ABSTRACT
Different similarity measures are available for comparison of textual data. These similarity measures are used for plagiarism detection. This research paper proposes a new similarity measure. Moreover, this paper proposes to consider length of content for plagiarism score determination.

General Terms
Data mining, plagiarism detection.

Keywords
Text data mining, plagiarism, similarity, information retrieval, web data mining .

1. INTRODUCTION
Use of internet is skyrocketing and people are availing every benefit from it. As the number of internet users is growing exponentially, fraudulent cyber activities are also increasing significantly. Copying contents from internet and submitting as one's own is a dishonest academic activity. Such unacknowledged copying of contents is called plagiarism. Academic plagiarism is rising in education and research sector. Many students practice the dishonest activity of plagiarism for submission of projects and assignments. It is not possible to verify genuineness of all the submitted assignments manually. Computerized system may be useful for detection of plagiarism. Different similarity measures are available for comparison of textual contents. This paper proposes a new measure for similarity analysis which would be helpful for plagiarism detection. It also proposes to consider length of the textual content in determination of plagiarism.

2. RELATED WORKS

Different research works are being carried out in the arena of similarity and plagiarism.

Christian collbarg, steven Koubhorov,Josua Louie and Thomas slattery, dept. of Computer Science, University of Arizona, Tuscan, AZ 85721 have developed a mechanism called SPLAT for determination of self plagiarism. Their system uses a WebL web spider that crawls through the web sites of the top fifty Computer Science departments, downloading research papers and grouping them by author. Next a text-comparison algorithm is used to search for instances of textual reuse. Instances of potential self- plagiarism for each author are reported in an HTML document so that they can be considered in more detail, in order to determine if they are truly self-plagiarized papers. The system discovered a number of pairs of papers of questionable originality.[1]

Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn and Supachanun Wanapu of Suranaree University of Technology, Thailand has proposed jaccard similarity measure for keyword matching in information retrieval. Technically, they developed a measure of similarity Jaccard with Prolog programming language to compare similarity between sets of data. Furthermore, the performance of this proposed similarity measurement method was accomplished by employing precision, recall, and F-measure. Precisely, the test results demonstrated the awareness of advantage and disadvantages of the measurement which were adapted and applied to a search for meaning by using Jaccard similarity coefficient. [2]

B. Karthikeyan, V. Vaithiyanathan, C. V. Lavanya of Sastra University, India have worked on similarity detection in source code. In a paper[3], they presented a study of three techniques, namely Jaccard Similarity (JS), Cosine Similarity (CS) and Jaccard Similarity with Shingles, with respect to source code plagiarism and compare the various results obtained. They have proposed a novel technique in which JS and CS are applied on the generated tokens from the parsed file, rather on the file directly. This helps in the fact that the files aren't considered just as plain text.

3. SIMILARITY MEASURES
Different similarity measures are available which can be used for comparison of textual contents. Following table illustrates some of the popular similarity measures along with the proposed measure:

Table1. Different Similarity Measures

Measure	Equation	Range
Jaccard	$J(x,y) = \frac{\lvert x \cap y \rvert}{\lvert x \cup y \rvert}$	0 to 1
Dice	$D(x,y) = \frac{2\lvert x \cap y \rvert}{\lvert x \cup y \rvert}$	0 to 2
Cosine	$Cos(x,y) = \frac{\sum_i (Xi, Yi)}{\sqrt{\sum_i (Xi)^2}\sqrt{\sum_i (Yi)^2}}$	0 to 1
Matching coefficient	M (x, y)=|x| − |x−y|	0 to |x| where |x|=|y|
Proposed	$\mathrm{p(x,y)} = 1 - \frac{\lvert \mathrm{x-y} \rvert}{\lvert \mathrm{x} \rvert}$	0 to 1

The proposed similarity measure listed in the last row of the above table yields better result in comparison to the other measures. To illustrate this, Let us consider the following two contents:
Content1: Cache memory has maximum hit value of one. [8 words]
Content2: Cache memory has minimum miss value with zero. [8 words]
Total unique words in these two contents is 12. Content1 has 4 words similar to content2 . As 4 words out of 8 words of content1 are same as content2's words, it can be said that Content1 is 50% similar with content2.

3.1 Jaccard Similarity:

S= |X ∩ Y| / |X U Y|

Let us check similarity of content1 and content2 using Jaccard similarity.

X={ Cache, memory, has, maximum, hit, value, of, one}

Y={ Cache, memory, has, minimum, miss, value, with, zero}
X∩Y ={ Cache, memory, has, value}

XUY = { Cache, memory, has, maximum, hit, value, of, one,minimum,miss,with,zero}
| X∩Y |=4 and |XUY|= 12
So, Jaccard Similarity
S= |A ∩B| / |AUB| =4/12= .33
It is seen that, although half of the total words were same, similarity percentage calculated using jaccard formula is **33% only**.

3.2 Dice's Similarity:

S= 2×|A ∩B| / |AUB|
For the contents, dice similarity is 2 × .33 = .66
Here it is seen that, although half of the total words were same, similarity percentage calculated using Dice's formula is **66%.**

3.3 Cosine Similarity:

For cosine similarity determination, following word set is constructed-

{Cache, memory, has, maximum, hit, value, of, one, minimum, miss, with, zero}

Numeric representation of content1 is {1,1,1,1,1,1,1,1,0,0,0,0}

Numeric representation of content2 is {1,1,1,0,0,1,0,0,1,1,1,1}

Cos(content1,content2)=4/√8.√8=4/7.95=.503. Hence, cosine similarity yields 50.3% similarity.

3.4 Matching coefficient:

For the example contents, matching coefficient is 8-4= 4. But as the maximum similarity value is not within any fixed range, this measure is not normalized.

3.5 Proposed approach:

In this paper, a new formula for similarity detection is proposed, which is-

$$p(x,y) = 1 - \frac{|x-y|}{|x|}$$

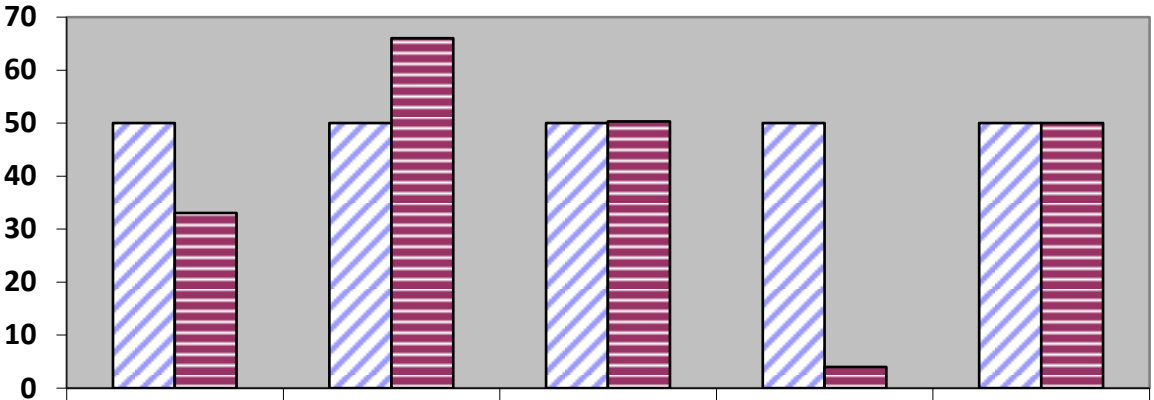
Considering the test contents-

|X-Y|= 4 . This is signifying number of words which are in X but not in Y. Content1 has 4 words {maximum, hit, of, one} which are absent in content2.

|X|=8.

So, similarity factor, S=1-4/8 = .5. This exhibits that content1 is **50%** similar to content2.So, it may be concluded that the proposed similarity measure yields better result in comparison to the other measures. Different commercial plagiarism detection tools are developed based on the measures discussed above. Comparative representation of the existing and proposed measures with graph and data are shown in table 2.

Table 2. Comparative result of different measures



	Jacard	Dice	Cosine	Matching Coefficient	Proposed
Actual	50	50	50	50	50
Calculated	33	66	50.3	4	50

4. IMPLEMENTATION

PL/SQL programming was done for practical implementation of the proposed similarity. The following function shown in TABLE 3 is created in ORACLE 10g which is callable by trigger. Triggers are programs which are executed automatically whenever a DML statement is performed.

Table 3. PL/SQL program

```
create or replace function proposed_similarity(var_c_id1 number,var_c_id2 number)
return number
is
var_wd_cnt_minus number(5):=0;  -- to hold total no of word data after minus operation
var_wd_cnt_frst_content number(5);  -- to hold total no of word in the first content
var_wd_frm_cur_minus varchar2(100);
var_simil_perc number(3);

-- following cursor is declared to hold minus result of the word data of two contents

 cursor word_data_cursor_minus is
    select word_data from wordtable where c_id=var_c_id1 minus
   select word_data from wordtable where c_id=var_c_id2;
begin

open word_data_cursor_minus;
loop
 fetch word_data_cursor_minus into var_wd_frm_cur_minus;
 exit when word_data_cursor_minus%notfound;
 var_wd_cnt_minus:=var_wd_cnt_minus+1;
end loop;
close word_data_cursor_minus;

select count(*) into var_wd_cnt_frst_content from wordtable where c_id=var_c_id1;
var_simil_perc:=(1-(var_wd_cnt_minus/var_wd_cnt_frst_content))*100;
return (var_simil_perc);
end;
/
```

A framework has been developed for plagiarism detection. The framework is developed using APEX(Application Express) and ORACLE 10g. Students projects, assignments etc. are submitted into the system. Each assignment is considered as a content and is stored in the CONTENT table. A trigger extracts individual words of the contents and are automatically inserted into the WORD_TABLE. Cardinality ratio between CONTENT and WORD_TABLE being 1:m, primary key C_ID is placed as foreign key in the WORDS_TABLE part.

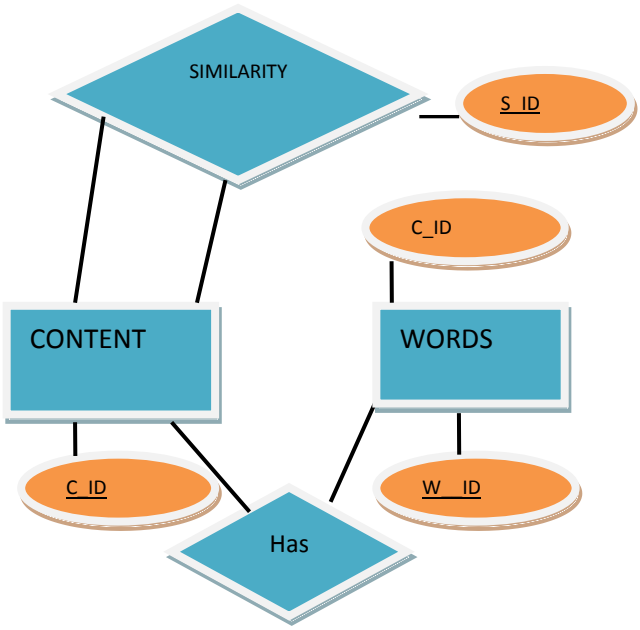


Figure 1: partial E-R diagram of developed plagiarism detection framework

For similarity calculation, a separate table SIMILARITY is maintained as shown in figure 1. This table stores similarity percentage between every pair of contents submitted into CONTENT table. Similarity is calculated by the developed function as shown in table3. This function is called by the trigger that fires on inserting data into CONTENT, that also performs the following necessary tasks-

- It removes beginning and ending spaces from a sentence.
- It converts multiple spaces present within a sentence into single space.
- It performs case changes for uniform comparisons.

```
SQL> @ proposed_similarity

Function created.

SQL> select similarity_function(1,5) from dual;

SIMILARITY_FUNCTION(1,5)
------------------------
                      20

SQL> select proposed_similarity(1,5) from dual;

PROPOSED_SIMILARITY(1,5)
------------------------
                      50

SQL> _
```

Figure 2: Testing of proposed similarity function

In the developed framework, each newly submitted content is checked against every content available in the database. Hence, N^{th} submission of content undergoes N-1 comparisons for similarity detection.

5. LENGTH CONSIDERATION

Similarity is not necessarily plagiarism. Two independently written short answers to a question can have many words in common. Only similarity should not be the deciding factor for plagiarism. 30 students were asked to write answer to a question within 8 words under strict invigilation confirming no plagiarism. The question was- " Write about the range of cache hit and miss ratio. [Strictly within 8 words]." Some sample answers are shown in table 4 below:

Table 4. Sample Short Answers

hit and miss range is o to 1.	Range of hit and miss is zero to one.	0 to 1 is the range of hit and miss.
Cache hit, miss range is 0 to 1.	hit and miss range is o to 1.	0 to 1 is the range of hit and miss.
hit and miss range is o to 1.	0 to 1 is the range of hit and miss.	hit and miss range is o to 1.
hit and miss range is zero to one.	Range of hit and miss is 0 to 1.	miss and hit range is o to 1.

Above answers are not at all copied from each other. But similarity measures show high similarity score amongst the answer contents.

Two long answers to a question, if share large number of words in common, may be potentially plagiarized. Hence, it is proposed that length of the content should also be considered for plagiarism calculation. Short contents, although similar, may not be plagiarized. Long contents, if similar, probability of plagiarism is more. In this paper, it is proposed :-

Length Factor, $LF= 1/ |X|^2$ where $|X|$ is the size of the content.

Plagiarism, P= S-LF if S>LF, otherwise P=S.

S is similarity.

Long answers, for example, answers with hundreds of words have nominal length factor, hence, similarity in words reflects plagiarism.

6. CONCLUSION

In this paper, a framework for plagiarism detection is described. Moreover, demonstration of proposed similarity measure and its accuracy over the other measures is explained. It is also proposed to consider length of content for deciding plagiarism score. This framework is capable of detecting literal plagiarism wherein plagiarists do not spend much time in hiding the academic crime they committed.[4] However, development of framework for detection of intelligent plagiarism[4] is set as future work.

7. REFERENCES

[1] "SPLAT:A system for self plagiarism detection" by Christian collbarg, steven Koubhorov,Josua Louie and Thomas slattery, dept. of Computer Science, University of Arizona, Tuscan, AZ 85721

[2] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn and Supachanun Wanapu, "Using of jaccard Coefficient for Keyword Similarity ", published in Proceedings of the International MultiConference of Engineers and Computer Scientists 2013 Vol I, IMECS 2013, March 13 - 15, 2013, Hong Kong

[3] B. Karthikeyan, V. Vaithiyanathan, C. V. Lavanya of Sastra University, India – "Similarity Detection in Source Code Using Data Mining Techniques" published in European Journal of Scientific Research ISSN 1450-216X Vol.62 No.4 (2011), pp. 500-505 © EuroJournals Publishing, Inc. 2011

[4] Salha M. Alzahrani, Naomie Salim, and Ajith Abraham, Senior Member, IEEE , Understanding Plagiarism Linguistic Patterns,Textual Features, and Detection Methods, published in Ieee Transactions On Systems, Man, And Cybernetics—Part C: Applications And Reviews, Vol. 42, No. 2, March 2012

[5] Paul Clough, Department of Information Studies, University of Sheffield,UK, "Old and new challenges in automatic plagiarism detection"

[6] www.oracle.com