iPlag: Intelligent Plagiarism Reasoner in Scientific Publications

Salha Alzahrani
Dept. of Comp. Science
Taif University
Taif, Saudi Arabia
s.zahrani@tu.edu.sa

Naomie Salim
Faculty of CS & Info. Sys.
Uni. Teknologi Malaysia
Johor, Malaysia
naomie@utm.my

Ajith Abraham
VSB Tech. Uni. of Ostrava
and MIR Labs
CZ and USA
ajith.abraham@ieee.org

Vasile Palade
Dept. of Comp. Science
Oxford University
United Kingdom
vasile.palade@cs.ox.ac.uk

***Abstract*— Existing anti-plagiarism tools are, in fact, text matching systems but do not make accurate judgments about plagiarism. Texts that are acceptable to be redundant and texts that are cited properly are all highlighted as plagiarism, and the real decision of plagiarism is left up to the user. To reduce the human input and to give more reliance to automatic plagiarism detectors, we propose an Intelligent Plagiarism Reasoner (iPlag), which works by combining several analytical procedures. Scholarly documents under investigation are segmented into logical tree-structured representation using a procedure called D-SEGMENT. Statistical methods are utilised to assign numerical weights to structural components under a technique called C-WEIGHT. Relevance ranking (R-RANK) and plagiarism screening approaches (P-SCREEN) are adjusted to incorporate structural weights, citation evidences, syntax-based and semantic-based methods into plagiarism detection results. We encourage current plagiarism detection systems to adapt the proposed framework.**

Keywords-intelligent reasoner; plagiarism detection; semantic; scientific publications

I. INTRODUCTION

Plagiarism is an academic misconduct; it copies others' contributions, hurts their feelings, and does not honour originators of ideas. It also rewards the offenders with marks or degrees they may not deserve. Plagiarism in *scientific publications* has increased, and it is becoming more possible that one day you will see your published work is used in another publication yet without proper attribution. Citation is a way to acknowledge previous works and to distinguish your contributions to the knowledge.

Scientific publications in the same field/area usually share the same general information. Besides, each publication should convey a specific message that contributes to that field. Different contributions can be made in different areas; for example, solving new problems, suggesting solutions to existing problems, experimenting different methods, comparing current methods, enhancing results, and so on. Such contributions of others are considered their ideas and should be acknowledged when reported in a further research. Plagiarism from scientific publications, hence, takes two forms: *words* and *ideas,* based on the *plagiarised content* [1]. Words plagiarism is the form of using some sentences or phrases found in other papers which contain some general knowledge but do not contain original ideas of others. Idea plagiarism, on the other hand, is the form of using others' innovative, inventive and original ideas as your own. Plagiarism in scientific publications has also been classified into: *literal* and *intelligent,* based on the *plagiarist's behaviour* [2]. In this regard, *literal* is the practice of verbatim or "cut and paste" of the text from one document to another (might be *words* or *ideas*). *Intelligent* practice involves the act of deceiving readers by changing the plagiarised text to appear in a different shape. Examples include excessive paraphrasing, comprehensive summarising and other substantial linguistic changes in the text. A taxonomy of plagiarism was proposed in [2] to classify various practices of plagiarism, as can be seen in Figure 1.

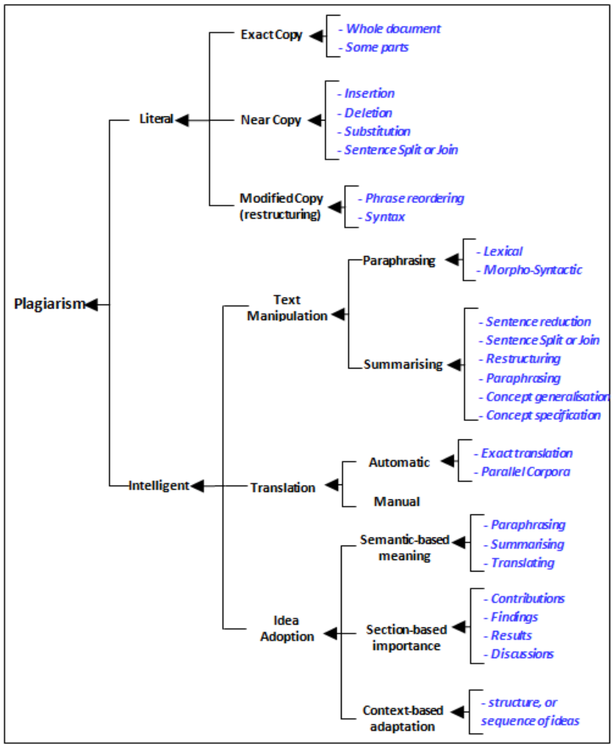

Figure 1. Taxonomy of plagiarism [2]

Different plagiarism tools have been surveyed in multiple works [3, 4]. Many online solutions are established today; for example, CrossCheck [5, 6], Turnitin [7], SafeAssign [8], EVE2 [9], WCopyFind [10], Viber [11], Scriptum [12], PlagiarismDetect [13], SCAM [14], CHECK [14], PPChecker [15], SNITCH [16], Ferret [17] and others. Several studies have reported that using plagiarism checker tools in academia is effective in reducing the problem [18], and discouraging the students

978-1-4673-0125-1 ©2011 IEEE

to commit plagiarism [19, 20]. Besides, anti-plagiarism tools help to educate students and authors in different disciplines about plagiarism [21]. Plagiarism detectors work to compare a submitted (i.e., query) document against an intra- or inter-corpus archives.

Methods developed in plagiarism detectors often rely on (i) exact string matching, or (ii) string similarity measures applied on fingerprints, chunks, sentences, or combinations. Such methods have been efficiently implemented to reveal *literal* plagiarism [2]. Fewer research works have been done [2] based on semantic relatedness, semantic variations, context adaptation, content reduction, and other methods which can be used to detect *intelligent* practices of plagiarism. To the best of our knowledge, plagiarism detection programs available today do not use any semantic-based methods.

In this paper, we propose a framework called *an intelligent plagiarism reasoner* (iPlag) that can be a useful addition (or enhancement) to existing systems. The framework defines an analytical procedure for analysing scientific publications, suspecting different kinds of plagiarism, highlighting significant instances, and referencing similar patterns in other publications. Sequential procedures are suggested to collaborate in detecting different types of plagiarism at the document level, section level, paragraph level and sentence level.

The rest of this paper is organised as follows. Section II discusses the related literature on plagiarism detection. Section III discusses the objectives of iPlag. Section IV explores the framework of iPlag and the methodology of its sub-systems. In section V, we discuss the features and the potential market of iPlag, and we draw a conclusion of this work in the last section.

II. Related Works

Research works on plagiarism detection will be discussed in this section in relation to the type of plagiarism to be detected (see Figure.1 for the taxonomy of plagiarism).

Methods to detect *literal* plagiarism involve duplicate/near-duplicate document detection to find documents that have been plagiarised in whole or in part [22, 23]. *Exact* string matching techniques have been applied widely to detect copied paragraphs, sentences and long sequences, such as word *n*-grams [24, 25]. Besides, string *similarity* gauging approaches have been implemented to detect plagiarism by inserting, deleting, and substituting phrases/words [26, 27]. These methods are also suitable to detect slight changes in the syntax of the text such as changing from active to passive forms. In [28], the Longest Common Sequence (LCS) similarity measure was combined with syntactical Parts-Of-Speech (POS) features to find plagiarism. Other plagiarism detection methods generally work through tokenising the text, constructing word *n*-grams, and utilising vector similarity metrics [29, 30]. Such methods can positively detect plagiarism when it is done by sentence/phrase recording and syntax rebuilding of statements.

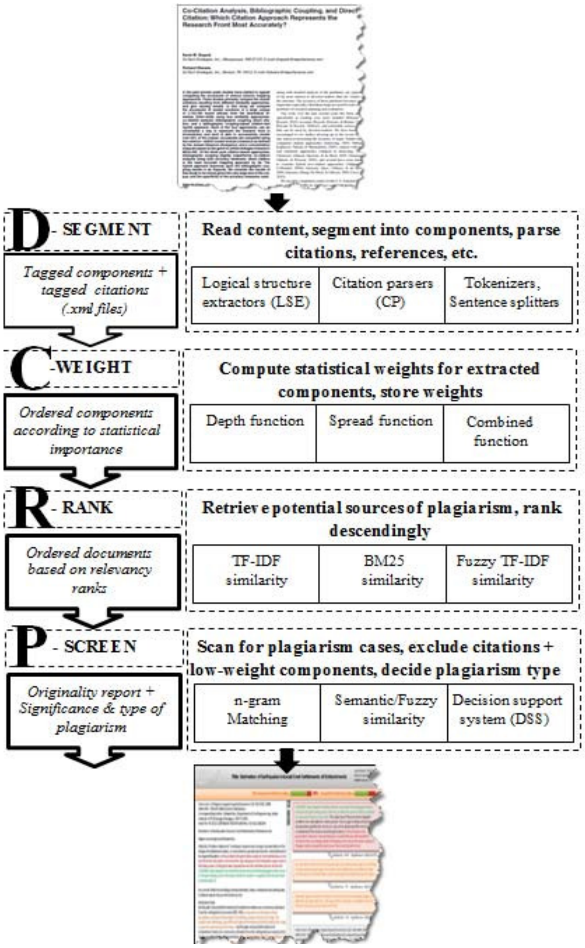
D - SEGMENT
Read content, segment into components, parse citations, references, etc.
Tagged components + tagged citations (.xml files)
Logical structure extractors (LSE)
Citation parsers (CP)
Tokenizers, Sentence splitters
C -WEIGHT
Compute statistical weights for extracted components, store weights
Ordered components according to statistical importance
Depth function
Spread function
Combined function
R - RANK
Retrieve potential sources of plagiarism, rank descendingly
Ordered documents based on relevancy ranks
TF-IDF similarity
BM25 similarity
Fuzzy TF-IDF similarity
P - SCREEN
Scan for plagiarism cases, exclude citations + low-weight components, decide plagiarism type
Originality report + Significance & type of plagiarism
n-gram Matching
Semantic/Fuzzy similarity
Decision support system (DSS)

Figure 2. Framework of Intelligent Plagiarism Reasoner (IPR)

Techniques to detect *intelligent* practices of plagiarism have gained less attention in the literature as reported in [2]. Particularly, plagiarism by paraphrasing the text or summarizing the content is not catered by the methods discussed above. In [31], the semantic similarity among short passages is computed based on lexical databases, word sequences, and corpus statistics. A fuzzy-based method was proposed to detect similar, but not necessarily the same, sentences [32]. Semantic-based and fuzzy-based methods can be successful to detect plagiarism by text manipulations. To detect translated plagiarism, a number of cross-language plagiarism detection approaches have been proposed [33-36]. A recent work [37] has discussed the use of structural information in scientific publications to detect significant plagiarism cases which, in fact, contributes to the concept of idea plagiarism (see Figure 1; Idea Adoption). This work is a complement and supplement to the work in [37]. Here we propose a framework using a more detailed representation of documents and a different plagiarism screening methods.

III. Problem and Questions

In this paper, we address the problem of analysing scientific publications against a set of source collection to detect plagiarism. We aim to explore some questions such as: (i) How to detect different types of plagiarism beyond verbatim plagiarism? (ii) How to report noteworthy plagiarism cases, deal with texts that are cited properly in a manuscript, and ignore texts acceptable to be redundant amongst several papers (e.g., acknowledgements)? (iii) How to detect plagiarised texts which have the same semantic meaning, but not the same words, with original sources?

IV. Framework of Intelligent Plagiarism Reasoner (iPlag)

This paper proposes a framework called *intelligent plagiarism reasoner* (iPlag) to report plagiarism in scientific publications that is accurate and reliable. The word "intelligent" refers to the detection method which is designed to simulate a human's checking of plagiarism, particularly in scholarly papers. Based on experience, one who is a professional in a certain field may check the originality of a scientific publication and find out about plagiarism manually. To clarify, an expert is able to: (i) decide the contributions in different parts of a paper such as findings, discussions and conclusions, (ii) suspect ideas or findings that might be taken from somewhere else, (iii) search the most relevant articles (e.g. having similar titles or abstracts). Meanwhile, the human checker does not pay attention to every statement but only sentences that convey original ideas, and ignores unimportant parts (e.g., general information, known theories, acknowledgements, etc.). The word "reasoner" refers to the ability of the framework to detect different kinds of plagiarism and to report which practice(s) was(were) used by the plagiarist to commit plagiarism. iPlag should have the capability to calculate the *degree of change* with regard to original texts, and the *degree of significance* in committed plagiarisms.

Figure 2 presents the framework of iPlag with four sequential procedures as follows. D-SEGMENT refers to the segmentation process of scientific publications into several structural components. A combination of statistical measures is used to assign a numerical weight (i.e. degree of importance) to the structural components in scientific publications via a procedure called C-WEIGHT. The outcome of the previous procedure is used to improve the retrieval results of relevant sources of plagiarism via the R-RANK module. Plagiarism screening process, called P-SCREEN, is applied based on a hierarchy of plagiarism detection methods wherein component weights, citation evidences, syntax correlation, and semantic relatedness are exploited to reason each plagiarism case. Figure 2 shows the possible outcomes from each stage. The following subsections discuss each procedure in detail.

A. Document Segmentation: D-SEGMENT

The framework starts with dividing scientific publications into several meaningful components. Then, a hierarchical tree-structured document representation is built to arrange the components in each document. Examples of trees include:

- `document->sections->paragraphs`
- `document->paragraphs->sentences`
- `document->topics->paragraphs` and so on.

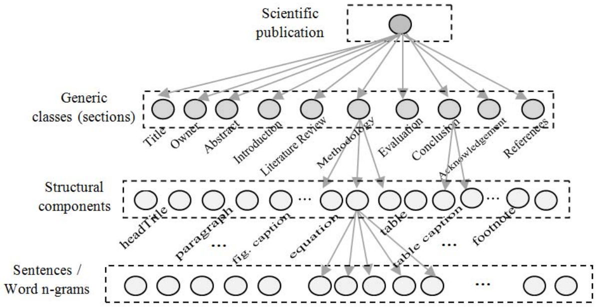

Figure 3. A 4-level hierarchical tree-structured representation of scientific publications

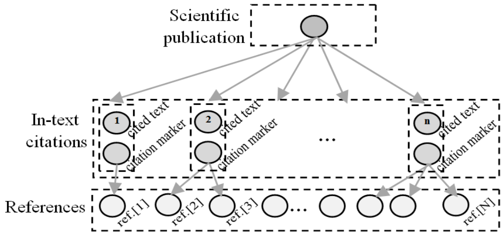

Figure 4. A 3-level citation evidence tree of scientific publications

In this framework, we use the logical structure organisation of scientific publications which include most (if not all) of the following categories: Title, Owner (i.e. author), Abstract, Introduction, Literature review (related previous works, etc.), Methodology, Evaluation (results, discussions, findings, etc.), Acknowledgements, and References. These categories are called *generic classes* and are represented as the 2nd level in the tree as shown in Figure 3.

Further, generic classes involve different structural *components* such as *head titles, paragraphs, tables, table captions, equations, figure captions, footnotes*, etc. Structural components are further segmented into sentences using a sentence splitter, and all texts are tokenised (i.e. divided into terms). As a result, scientific publications are represented as a four-level tree:

- `document->classes->components->sentences`

Such comprehensive tree-structured representation, shown in Figure 3, assists to understand the topics and the ideas in the publication.

Citation parsers, such as the tool developed by Loung et al. [38], can be applied to extract citations and references. Figure 4 shows a three-level tree for parsed citations in scholarly documents. The tree is represented as follows:

- `document->inTextCitation(text,marker)->references`

whereby the in-text citation has two elements. First is the fragment of text (quoted or rephrased) that is taken from another reference. Second is the citation marker which links that piece of text to one of the references. Citation markers take two forms: numerical (e.g. IEEE style in this paper), and authorial (e.g. writing the author's last name).

B. Component Weighting: C-WEIGHT

Structural components within an article can be assigned numerical weights to indicate how important that component is to the article. In [37], we explored the use of structural information to detect significant plagiarism cases in scientific publications. Different functions (eleven equations) were proposed to compute components' weights based on three statistical measures namely *Inverse Generic Class Frequency*, *Depth* and *Spread*. The study compared these functions and reported that combined Spread and Depth improved the retrieval of potential candidates to the suspected document and yielded better detection results [37].

Based on the above, we suggest the use of statistical measures as follows. *Spread* of a term *t* defines the number of structural components that contain the term. *Depth* of a term *t* refers to the frequency of the term in a class (unlike normal term frequency which considers the number of the term's occurrences in the whole document). Thus, *Spread* and *Depth* can be combined to find the weight *w* of a structural component *c* in a generic class *Gc*, which can be expressed as follows:

$$w(c)=\sum_{t\in c} spread(t)\times\frac{tf_{t,Gc}}{\max tf_{Gc}} \tag{1}$$

where *spread*(*t*) is the number of components that has *t*, $tf_{t,Gc}$ is the frequency of *t* in the generic class *Gc* which has *t*, and $\max tf_{Gc}$ is the maximum frequency occurs in that class.

C. Relevance Ranking: R-RANK

Instead of comparing the submitted publication with everything in the source archives, relevant publications that may be the potential sources of plagiarism should be retrieved at glance. In iPlag, we recommend a method to order relevant candidates according to the weights of structural components. In this sense, a modification on typical retrieval methods could be done to consider weighted components. In [39], the authors adapted the original BM25 similarity ranking formula with regard to the weights defined in different blocks of webpage documents. The similarity of a document *d* and a user query *q* is expressed as follows:

$$sim(d,q)=\sum_{t\in d\wedge t\in q}\frac{(k+1)\times tf'_{t,d}}{k(1-l+l\frac{|d|}{|d'|})+tf'_{t,d}}\times\log\frac{|D|-|D_t|+0.5}{|D_t|+0.5} \tag{2}$$

where $|d|$ refers to the document's length (i.e. number of terms in *d*), $|d'|$ is all documents average length, $|D|$ is the corpus size (i.e. number of documents in the collection), and $|D_t|$ is the number of documents containing *t*. Notice that *k* and *l* are parameters which can be adjusted to a particular corpus. Further, $tf'_{t,d}$ is a modified term frequency of *t* in the document *d* which includes the block's weight in web documents (likewise component's weight in scientific publications), given as:

$$tf'_{t,d}=\sum_{block\in d} tf_{block}\times w_{block} \tag{3}$$

Similarly in [37], we modified the term frequency – inverse document frequency (TF-IDF) weighting used in typical information retrieval systems to comprise the weights of structural components. Thus, the similarity formula of a document *d* with regard to a user query *q* is given by:

$$sim(d,q)=\frac{\sum_{t\in d\wedge t\in q} w_{t,d}\times w_{t,q}}{\sqrt{\sum_{t\in d\wedge t\in q}\left(w_{t,d}\right)^2}\times\sqrt{\sum_{t\in d\wedge t\in q}\left(w_{t,q}\right)^2}} \tag{4}$$

where $w_{t,d}$ and $w_{t,q}$ is expressed based on term frequency $tf_{t,c}$ of a term *t* in a component *c*, and the weight *w*(*c*) as expressed in (5).

$$w_{t,d}=\sum_{G\in d}\sum_{c\in G} tf_{t,c}\times w(c).\log\frac{|D|}{|D_t|} \tag{5}$$

Notice that *d* refers to the documents in the source archives while *q* refers to the submitted document in the plagiarism detection systems. Documents that gain higher relevancy (i.e. *sim*(*d*,*q*) based on either formula) than a *threshold* α will be retrieved and ranked. The threshold can be tuned during the experiments to a particular collection.

D. Plagiarism Screening: P-SCREEN

Retrieval of a list of relevant sources is not the end goal of any plagiarism detection system. Users normally wish to see side-by-side comparisons of submitted documents and their candidates. The last part of the framework involves several sub-procedures which corporately aim to scan the submitted publication for plagiarism cases and present the originality report.

In [37], we examined a structural component-based overlapping between the submitted publication's components and all relevant sources. The overlapping distance between two components *c* and c' was suggested as follows [37]:

$$Overlap(c,c')=\Delta\cdot\left[\frac{|c\cap c'|}{|c\cup c'|}\right] \tag{6}$$

where Δ is the significance factor as will be explained shortly. The formula in (6) suggests a small distance if both components share many of their terms, and a big distance (i.e. dissimilarity) otherwise. This method is, in fact, successful to report verbatim plagiarism, or when sentences are restructured and phrases are reordered. Notice that the method works on the component-level (e.g. paragraphs). Although this approach may be faster than statement-by-statement comparison, recall results principally need to be improved (highest recall≈0.6).

To enhance plagiarism detection results, we propose a plagiarism screening procedure that is capable to handle different types of plagiarism with accuracy and reliability. The procedure is "somehow" similar to a decision support system (DSS) in the sense that it aids to decide which type of plagiarism, and how does it matter based on known factors found in the text.

To start with, we refer to the submitted publication (i.e., the query document) as *q*, and the source publication ranked as relevant as *d*. In the first step, all components in *q* and *d* with weight *w*(*c*) equals (or nearly equals) to zero are excluded. This is to filter out sections that is not important to be checked such as acknowledgement, references, authors' bio-data, etc. Further, components in *q* with citation evidence are completely excluded. Texts within a component that have citation evidence are also removed (but not the whole component). Remaining texts will be analysed in the next steps. These steps may reduce texts to be processed with an approximate ratio of 10%-30%.

Then, the overlapping distance (i.e. Jaccard similarity) between components in *q* and *d* (3rd-level nodes) is computed according to equation (7), and a plagiarism case will be regarded if most of the text in *q*'s component is overlapped with *d*'s component. A decision can be given for these plagiarism cases as "literal plagiarism".

$$Overlap(c,c')=\left[\frac{|c\cap c'|}{|c\cup c'|}\right] \tag{7}$$

where $|c\cap c'|$ is the number of common terms between components c and c', and $|c\cup c'|$ is the total number of terms in both components. Otherwise, an additional method will be carried out based on the 4th-level nodes. We suggest using a syntax-based approach and, if no plagiarism is detected, a semantic-based approach is used to find sentences (or word n-gram sequences) that are semantically, but not literally, plagiarised. Methods such as the one proposed in [40] should suffice for this purpose. If none of the methods reports plagiarism, the text is described as "plagiarism-free".

Adjacent sentences (or word n-grams) regarded as plagiarism are joined together, and the final plagiarism cases are reported along with the decision that indicates the type of plagiarism. If a syntax-based approach is used to detect successive plagiarised sentences (or word n-grams), the type is "modified copy". Nevertheless, if semantic-based (or both) approaches are used to detect plagiarism, then a "paraphrased plagiarism" is decided here. Advanced methods could be applied to detect "summarised plagiarism" and "translated plagiarism" but they are not the focus of this framework.

The originality report is the final outcome. In addition to the side-by-side presentation of submitted query and candidate sources, we suggest to present the following parameters.

Definition 1: The *degree of change* in a detected plagiarism case *p* is defined as the number of terms changed with regard to the original text. It can be given by the equation:

$$cha(p)=\frac{|T_p|-|T_p\cap T_o|}{|T_p\cap T_o|}\times 100 \tag{8}$$

where T_p is the length (i.e. number of terms) of a plagiarism case *p*, and T_o is the length of the original text in the source document.

Definition 2: The *degree of significance* in a detected plagiarism case *p* is defined as the multiplication of the components' weights that hold the plagiarised text and the original text. It can be expressed as follows:

$$sig(p)=\Delta=w(c)\times w(c') \tag{9}$$

where *c* is the component that contains *p* in the query document, c' refers to the component that has the original text, and *w*(*c*) is defined in equation (1).

Both degrees of significance and change are defined with regard to each plagiarism case. Another two parameters can be defined on the document-level.

Definition 3: The *similarity index* (*SI*) of a submitted publication *q* and a source publication *d* is the length of all plagiarism cases found in *q* that are taken from the original document d, as follows:

$$SI(q,d)=\frac{\sum|p:p\in q\wedge p\in d|}{|q|}\times 100 \tag{10}$$

where |*q*| is the total length (in terms) of *q*, and |*p*| is length of a plagiarism case in *q* plagiarized from *d*.

Definition 4: The *overall similarity index* (*OSI*) of *q* is the percentage of all plagiarism cases found in *q*.

$$OSI(q)=\frac{\sum|p:p\in q|}{|q|}\times 100 \tag{11}$$

The overall plagiarism screening process is summarized in the next pseudo code:

```
iPlag: Pseudo code for plagiarism screening
Input: query document q
Input: source collection archive $D=\{d_1,d_2,\dots,d_N\}$
Input: relevant source documents
                   $D'=\{d_1,d_2,\dots,d_n\}$: D'CD, n<N

START
  For Each d in D'
    Retrieve d's tree from database
    Sort components based on weights w(c)
    Remove components with w(c)≈0
  Next d

  Read q
  Build document tree using LSE tool
  Build citation tree using CP tool
  Sort components based on weights w(c)
  Remove components with w(c)≈0

  For Each component c
    If all text in c with citation evidence?
      Remove c
    Else If some text in c with citation evidence?
      Remove cited text
    For Each d'
      For Each c'
        Compute overlap(c,c')
        If overlap≈≈1
          Decision = "literal plagiarism"
          Add P(c,c')
        Else If syntax_sim(c,c')≥threshold
          Decision = "modified copy"
          Add P(c,c')
```

iPlag: Pseudo code for plagiarism screening (cont.)

```
    Else If symantic_sim(c,c')≥threshold
     Decision = "paraphrased plagiarism"
     Add P(c,c')
    Else
     Decision = "plagiarism-free"
     Add P(c,c')
   Next c'
  Next d'

  Print Decision, P(c,c')
  Compute degree-of-significance
  Compute degree-of-change
 Next c
 Compute SI, OSI
END
```

V. Features and Potential Market

Important features that characterise the system is partitioning the text into semantic organisation and building citation evidence tree extraction using existing well-developed free tools. Other features include component weighting for important contents and advanced plagiarism screening results, such as *degree of significance* and *degree of change* in detected instances. Potential market for the proposed framework include patent offices, educational providers, universities, schools, publishers and archive managers who wish to use offline/online plagiarism detection systems. It can also be used in academic writing assistance software.

VI. Conclusion and Future Work

This paper describes a conceptual framework for a plagiarism reasoner called iPlag that deems to give "intelligent" decisions on plagiarism practices found in scientific publications. The framework is described in detail throughout the paper and we believe that it is possible to adapt current plagiarism detection tools to use the iPlag's features. As this paper focused on the development issues mainly, future work will include the evaluation of this framework and comparison with existing plagiarism detection systems.

Acknowledgment

This first author is sponsored by Taif University, Kingdom of Saudi Arabia. This research is sponsored partly by the Research University Grant of Science and Technology Vote Number: 79302, Universiti Teknologi Malaysia.

References

[1] M. Bouville, "Plagiarism: Words and ideas," *Science & Engineering Ethics,* vol. 14, pp. 311-322, 2008.

[2] S. M. Alzahrani, N. Salim, and A. Abraham, "Understanding Plagiarism Linguistic Patterns, Textual Features and Detection Methods," in *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*. vol. PP, 2011, pp. 1-17 [online].

[3] H. Maurer, F. Kappe, and B. Zaka, "Plagiarism - A Survey," *J. Universal Computer Science,* vol. 12, pp. 1050-1084, 2006.

[4] L. Romans, G. Vita, and G. Janis, "Computer-based plagiarism detection methods and tools: an overview," in *Proc. Int. Conf. Comp. Systems Technologies*, Bulgaria, 2007.

[5] H. Zhang, "CrossCheck: an effective tool for detecting plagiarism," *Learned Publishing,* vol. 23, pp. 9-14, 2010.

[6] K. Meddings, "Credit where credit's due: plagiarism screening in scholarly publishing," *Learned Publishing,* vol. 23, pp. 5-8, 2010.

[7] Turnitin. Available: http://www.turnitin.com

[8] SafeAssign by BlackBoard. Available: http://www.safeassign.com/

[9] EVE2: Available: http://www.canexus.com/eve/download.shtml

[10] WCopyFind. Available: http://www.plagiarism.phys.virginia.edu

[11] Viper, the plagiarism checker. Available: http://www.scanmyessay.com/

[12] April, 23 2008). Scriptum. Available: http://www.scriptum.ca.

[13] PlagiarismDetect. Available: http://www.plagiarismdetect.com/

[14] N. Shivakumar and H. Garcia-Molina, "SCAM: A Copy Detection Mechanism for Digital Documents," *D-Lib Magazine,* 1995.

[15] N. Kang, A. Gelbukh, and S. Han, "PPChecker: Plagiarism Pattern Checker in Document Copy Detection," in *Text, Speech and Dialogue*, 2006, pp. 661-667.

[16] N. Sebastian and P. W. Thomas, "SNITCH: a software tool for detecting cut and paste plagiarism," in *Proc. 37th SIGCSE Symposium on Computer Science Education*, New York, USA, 2006, pp. 51-55.

[17] P. C. R. Lane, C. Lyon, and J. A. Malcolm, "Demonstration of the Ferret plagiarism detector," *2nd Int. Plagiarism Conf.,* 2006.

[18] J. Chaudhuri, "Deterring digital plagiarism, how effective is the digital detection process?," *Webology,* vol. 5, 2008.

[19] I. Anderson, "Avoiding plagiarism in academic writing," *Nursing standard,* vol. 23, pp. 35-37, 2009.

[20] N. Beute, E. S. van Aswegen, and C. Winberg, "Avoiding Plagiarism in Contexts of Development and Change," *IEEE Trans. Education,* vol. 51, pp. 201-205, 2008.

[21] S. R. Whittle and D. G. Murdoch-Eaton, "Learning about plagiarism using Turnitin detection software," *Medical Education,* vol. 42, p. 528, 2008.

[22] M. Elhadi and A. Al-Tobi, "Duplicate detection in documents and webpages using improved longest common subsequence and documents syntactical structures," in *Proc. 4th Int. Conf. Computer Sciences and Convergence Information Technology*, Seoul, Korea, 2009, pp. 679-684.

[23] C. D. Manning, P. Raghavan, and H. Schütze, "Web search basics: Near-duplicates and shingling," in *Introduction to Information Retrieval*: Cambridge University Press, 2008, pp. 437-442.

[24] C. Grozea, C. Gehl, and M. Popescu, "ENCOPLOT: Pairwise sequence matching in linear time applied to plagiarism detection," in *Proc. SEPLN'09*, Donostia, Spain, 2009, pp. 10-18.

[25] C. Basile, D. Benedetto, E. Caglioti, G. Cristadoro, and M. D. Esposti, "A plagiarism detection procedure in three steps: Selection, Matches and "Squares"," in *Proc. SEPLN'09*, Donostia, Spain, 2009, pp. 19-23.

[26] V. Scherbinin and S. Butakov, "Using Microsoft SQL server platform for plagiarism detection," in *Proc. SEPLN'09*, Donostia, Spain, 2009, pp. 36-37.

[27] Z. Su, B. R. Ahn, K. Y. Eom, M. K. Kang, J. P. Kim, and M. K. Kim, "Plagiarism detection using the Levenshtein distance and Smith-Waterman algorithm," in *Proc. 3rd Int. Conf. Innovative Computing Information and Control*, Dalian, Liaoning, 2008.

[28] M. Elhadi and A. Al-Tobi, "Use of text syntactical structures in detection of document duplicates," in *Proc. 3rd Int. Conf. Digital Information Management*, London, UK, 2008, pp. 520-525.

[29] M. Murugesan, W. Jiang, C. Clifton, L. Si, and J. Vaidya, "Efficient privacy-preserving similar document detection," *VLDB Journal,* 2010.

[30] C. Lyon, J. A. Malcolm, and R. G. Dickerson, "Detecting short passages of similar text in large document collections," in *Proc. Conf. Empirical Methods Natural Language Processing*, 2001.

[31] Y. Li, D. McLean, Z. A. Bandar, J. D. O'Shea, and K. Crockett, "Sentence similarity based on semantic nets and corpus statistics," *IEEE Trans. Knowledge and Data Engineering,* vol. 18, pp. 1138-1150, 2006.

[32] R. Yerra and Y.-K. Ng, "A Sentence-Based Copy Detection Approach for Web Documents," in *Fuzzy Systems and Knowledge Discovery*, 2005, pp. 557-570.

[33] A. Barrón-Cedeño, "On the mono- and cross-language detection of text reuse and plagiarism," in *Proc. 33rd Annu. Int. ACM SIGIR*, Geneva, Switzerland, 2010.

[34] A. Barrón-Cedeño, P. Rosso, D. Pinto, and A. Juan, "On cross-lingual plagiarism analysis using a statistical model," in *Proc. ECAI'08 PAN Workshop*, Patras, Greece 2008, pp. 9-13.

[35] R. Corezola Pereira, V. Moreira, and R. Galante, "A New Approach for Cross-Language Plagiarism Analysis," in *Multilingual and Multimodal Information Access Evaluation*. vol. 6360, M. Agosti, N. Ferro, C. Peters, M. de Rijke, and A. Smeaton, Eds.: Springer Berlin / Heidelberg, 2010, pp. 15-26.

[36] M. Potthast, A. Barrón-Cedeño, B. Stein, and P. Rosso, "Cross-language plagiarism detection," *Lang. Resources & Evaluation,* vol. Online First, pp. 1-18, 2010.

[37] S. Alzahrani, N. Salim, and A. Abraham, "Using structural information and citation evidence to detect significant plagiarism cases," in *Journal of the American Society for Information Science and Technology (JASIST), [InPrint]*. 2011.

[38] M.-T. Luong, T. D. Nguyen, and M.-Y. Kan, "Logical structure recovery in scholarly articles with rich document features," *Int. J. Digital Library Systems,* vol. 1, pp. 1-23, 2010.

[39] E. S. de Moura, D. Fernandes, B. Ribeiro-Neto, A. S. da Silva, and M. A. Gonçalves, "Using structural information to improve search in Web collections," *J. Am. Soc. Inf. Sci. Technol.,* vol. 61, pp. 2503-2513, 2010.

[40] S. Alzahrani and N. Salim, "Fuzzy semantic-based string similarity for extrinsic plagiarism detection: Lab Report for PAN at CLEF'10," in *Proc. 4th Int. Workshop PAN-10*, Padua, Italy., 2010.