Towards Developing an Intelligent Plagiarism Detection System

Rayees Ahmad Dar
Dept. of Computer Science and Engineering, IUST

Dr. Assif Assad
Dept. of Computer Science and Engineering, IUST

ABSTRACT

Plagiarism is an academic offence. Though the software currently available can detect the simpler kinds of plagiarism like the copy paste however they are not able to detect other advanced types of plagiarism like paraphrasing, use of synonyms etc. In this work we employ the semantic role labelling (SRL) technique coupled with the pronoun resolution to detect this type of plagiarism. We tested our system on the short answers corpus and achieved respectable accuracy.

Keywords: Plagiarism, paraphrasing, semantic role labeling, pronoun resolution.

I. INTRODUCTION

Plagiarism is one of the key problems in the academic and scientific world where people use others' ideas and work and pose them as their own without acknowledgement. Perhaps one of famous examples of plagiarism was the Downing Street "dodgy dossier" [1]. The advent of internet and the subsequent easy availability of vast amounts of information have made the issue of detecting plagiarism a much more relevant challenge. Plagiarism detection involves the detection of passages from a suspect document that have been plagiarized from some source document [2]. Manual plagiarism is cumbersome and requires a vast amount of knowledge. Researchers have been trying to find automated methods to detect plagiarism in documents with much attention focused on the student plagiarism.

Different approaches have been proposed for the plagiarism detection based on the various text features. Some of the popular tools available for plagiarism checking like turnitin [1] (used by Pondicherry University as well) use methods based on simple string matching, n-gram matching, syntax analysis etc. Other than the semantic methods most of the other methods can be fooled by the plagiarist by simple text modification methods like synonym substitution and paraphrasing. However semantic methods that focus on the semantic features of the text are hard to be bypassed.

The use of semantic role labelling (SRL) in the plagiarism detection by Osman et al. [3], [4] has shown great promise. It uses the semantic roles of the various terms used in two sentences to compare them. The semantic roles provide us information about questions like who, what, where, how etc. by identifying the subject, object, verb and other entities in the sentence. Since the semantic roles do not change by simple paraphrasing, this renders the method useful for plagiarism detection. However there are various aspects which this method has overlooked. Hence the accuracy of the method can be greatly improved by certain additions and improvements to the basic technique proposed.

Anaphora resolution refers to finding the respective antecedent of an expression. In simple words, it refers to referencing back to the antecedent which provides us information about the anaphor. An anaphor could be a pronoun or a noun phrase. In the SRL based plagiarism detection method we actually get many false positives because of the lack of ability of the system to disambiguate the pronouns properly. In this work we would present a framework to incorporate the anaphora resolution into the SRL based plagiarism detection technique. We would also investigate the use of a proper similarity measure technique to account for normalization. Also the impact of the accuracies of various techniques used like SRL, anaphora resolution on the main method and suggest some improvements and solutions.

The rest of the paper is organized as follows: in section II a brief literature survey will be provided, section III will discuss the SRL based plagiarism detection method and its shortcomings, section IV will introduce the anaphora resolution and the proposed framework to incorporate it into the already proposed technique along with some improvements. Conclusion and future work will be discussed in section V.

II. LITERATURE SURVEY

The plagiarism detection can be broadly classified into two categories – internal and external. Internal plagiarism detection refers to the analysis of the writing style of a document to find any deviations in it so as to establish if a certain portion of the document is plagiarized. The interesting feature of this class of methods is that it involves analysis of only one document and no library of documents is needed. A lot of research has been done in this field [5].

The other class of detection methods which has received most attention from researchers is external plagiarism detection. In external detection a suspicious document is analyzed against a group of source documents to find portions of text from the suspicious document that have been plagiarized usually with reference to the document (mostly mentioning the paragraph) from which it was plagiarized.

There are a lot of methods that have been devised for the task of external plagiarism detection based on various comparison factors like textual features, syntactic features, semantic features and structural features of the text.

Researchers have been trying to exploit the semantic features of the text for plagiarism detection because they contain the intelligible information. Hence if we could devise methods that exploit such features then we could be able to detect most of the intelligent plagiarisms.

Li et al. [6] presented an algorithm for computing similarity between texts of sentence length by taking into account the semantic information and the word order. Information from a structured lexical database and from corpus statistics was used to compute the semantic similarity. The main contribution of this method was that it even considered the word order in the total string similarity.

In [7], Bao et al. proposed a semantic sequence kin which, based on the local semantic density, finds out the semantic sequences, representing locally the frequent semantic features, and then all

of the semantic sequences are collected to imply the global features of the text. Although complex, but it proved to a good choice for detecting reworded sentences.

In [8] Alzahrani et al. used a fuzzy method to find the similarity between sentences with the help of WordNet thesaurus. Hence it was able to detect plagiarisms involving synonyms usage but its precision and recall values were still very low.

Chow and Salim [9] proposed a different semantic method which calculates similarity between two documents according to the predicates in the sentences. Here again WordNet is employed to find the similarity between the predicates. However the drawback of this method is that not all the parts of the sentence are considered and similarity is computed based on verb, subject and object only.

All the above methods were still primitive until Osman et al. [3] used the semantic role labelling to label the different terms in the sentence with different semantic roles like verb, agent, patient, location etc. This way the total sematic structure of the sentence is a captured and if any tampering is done with the text it can be easily detected. The authors also studied the effect of various arguments to find important arguments in the similarity detection process, thereby the unimportant arguments were discarded.

Our work is based on the work done by Osman et al. [3] with the following important modification. Firstly since the pronouns are not resolved in the Osman method, it unnecessarily results in low similarity score. We use pronoun resolution to counter this shortcoming. We also note that unlike other arguments the agent (subject, A0) and patient (object, A1, A2) are more important. Hence we devise a precondition for the sentences to pass before calculating the final similarity score based on all the arguments. Our work has shown relatively better results than the basic Osman method.

III. Pronoun Resolution

Pronoun resolution is one of the most difficult tasks in natural language processing. In pronoun resolution the proper antecedent of a pronoun is found. In fact pronoun resolution is part of a much wider problem known as anaphora resolution where references to noun phrases (which could be a full-fledged noun phrase, a pronoun, a demonstrative or a reflexive) are resolved with their antecedents. It is also known as co-reference resolution although there is a slight difference between the two [10]. A lot of research has been done in this field based on various approaches like knowledge rich methods, including syntax based methods and discourse-based methods, and knowledge poor methods using statistical NLP techniques [11]. There are many tools available for anaphora resolution like Stanford CoreNLP co-reference module, Guitar, JavaRAP etc.

IV. System Design

The use of pronoun resolution coupled with SRL is aimed at detecting the semantic similarity between two sentences. In this section we will discuss the proposed method. We first apply the pronoun resolution to the source and suspected documents. This is applied at the start so that no information is lost because the pre-processing techniques result in omission of certain information. The pronoun resolution is followed by various pre-processing techniques like stemming, stop-word removal etc. After stripping the document of unnecessary details and simplifying the text, the SRL is applied to extract the semantic information in the form of arguments from the documents. The application of SRL ultimately results in the creation of verb trees, where each verb in the document becomes a parent and the others arguments (agent, patient etc.) become the children. Here it must be noted that a verb tree can itself be a child of some other verb tree, particularly in case of complex sentences where sub-clauses are present.

Fig. 1 demonstrates the general architecture of the proposed method with the various stages. It must be noted that once the various stages are applied to the various documents in the repository it shall be easy to store the document as verb-trees. Hence whenever a comparison is needed its verb-tree structure will be extracted.

A. Pronoun Resolution

This is applied at the beginning so that the resolution is more accurate. In this stage we replace all the pronouns with their respective antecedents. An example of such a process is given in Fig. 6 in appendix I. In the example the pronoun **He** is replaced by its proper noun **John**.

B. Text Segmentation

Text segmentation refers to the chunking of text into various units like paragraphs, sentences, clauses etc. In our case we consider sentence as a basic unit of information and the plagiarism considered is assumed to be at the sentence level.

C. Stop words removal and stemming

Stop words refer to the various words in the text which are used for decoration purposes, punctuation, connectives, articles etc. These words are not semantically much needed hence to simplify the computation should be removed without much loss of semantics. This is particularly important because for complex sentences we might want to analyze individual clauses separately because some plagiarists would break a complex sentence and then in the absence of this process we might not be able to detect the plagiarism.

Stemming is somewhat alike process where root words are found. Therefore it results in dropping of suffixes and prefixes. Also verbs are brought to the basic forms known as lemmas. This might be detrimental in certain cases. For example all the forms of **be** verb like **is**, **was**, **were** are converted to **be**, therefore any deeper difference is lost. However it is useful in other cases where the plagiarist merely changes the way of narration from one person, say first person, to other person, say third person, or changes the tense particularly in literature this

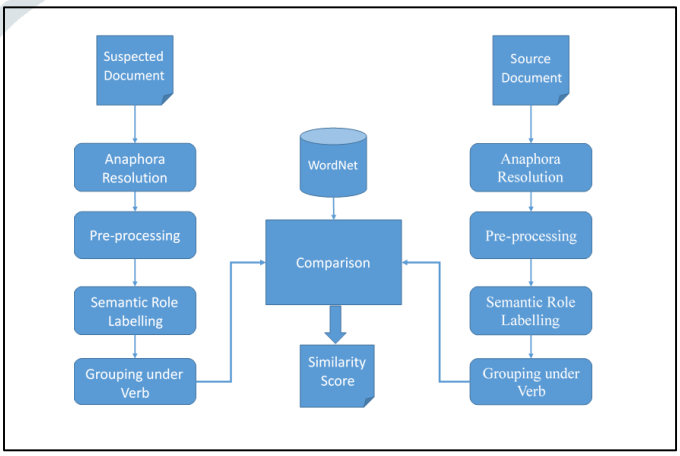


Fig. 1. System model

is a common case. It must be noted that the pre-processing results in loss of some information but since the same pre-

processing is applied in parallel to both the documents the effect is mostly cancelled out. Hence we can safely ignore the side-effects given the advantage of much simplification of the text.

D. Sematic Role Labelling

Here the pre-processed text is taken and SRL applied to extract the various arguments. The arguments role of each term in the sentence. An example is given in Fig. 7 in appendix I. Fig. 9 in appendix I gives the meaning of the various argument labels.

E. Verb – tree formation

Here each verb in the sentence is taken and a tree is formed with the various arguments as its children. This is particularly important because it helps to organize the document as a conceptual entity. In case of sentences made of sub-clauses, the individual clauses also make individual trees (if a verb is present) and then the main verb in the sentence takes these sub-clause verb-trees as its children. In case a plagiarist breaks a sentences into smaller sentences, the presence of individual sub-clause verb-trees would capture it easily. Fig. 8 in appendix I demonstrates the formation verb-tree for a sentence.

F. Comparison

This is the last and the most important stage where the actual comparison between the documents is done and possible cases of plagiarism are found out. Algorithm 1 gives the actual process of comparison. The algorithm is supplied with pre-processed documents up to the stage of SRL (see Fig. 1). As a first step the verb-trees of both the documents are constructed like the one shown in Fig. 8 in appendix I. After constructing the verb-trees the documents are matched sentence by sentence to find any semantic similarity. The similarity between two sentences is found using algorithm 2 which returns a similarity value θ. A case of plagiarism is established if the value of θ is greater or equal to a threshold value θ^T. The value of θ^T determines the amount of tolerance to the variation of documents. In our case we have set it to 0.6.

In algorithm 2, the verbs in the sentences are matched for similarity using a similarity function *isSimilar()* which checks if two strings are same or not. Here string s1 is similar to string *s2* if *s1* is same as *s2* or *s1* is in the synset of *s2*. Synset is the set of synonyms of a word obtained from the WordNet thesaurus [12]. Whenever a match occurs the verbs are tested to pass the minimum criteria devised. The criteria stems from the fact that if two sentences are similar then their subject and object arguments should match necessarily. However it is not necessary that both sentences contain both the arguments. Lines … account for this fact. *Arg_sim()*, given in Algorithm 3 is used in algorithm 2 for checking the criteria. If the verbs pass the criteria, then a similarity score is calculated based on the cosine function *co_sim()* given by (1). An illustration of the actual process is given in appendix IV.

Given two vectors *A* and *B* representing the two sentences, the cosine similarity *co_sim* is calculated as in (1).

$$co\_sim = \frac{\vec{A}.\vec{B}}{\|\vec{A}\|\|\vec{B}\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2}\sqrt{\sum_{i=1}^{n}(B_i)^2}} \quad (1)$$

Algorithm 1: *document_comparison*
Input: Preprocessed documents **d1** and **d2**
Output: Prints the sentenced detected to be plagiarized.

```
1.  For all sentences si in d1
2.    For all sentences sj in d2
3.      θ = sentence_comparison(si, sj)
4.      If (θ ≥ θ^T)  //θ^T is the threshold score
5.        Print(si,sj)
6.      Endif
7.    Endfor
```

Algorithm 2: *sentence_comparison*
Input: Sentences **s1** and **s2**
Output: Similarity score between **s1** and **s2**
// **A0 and A1** stand for the argument labels of
//subject and object, respectively.

```
1.  For all verbs vi in s1
2.    For all verbs vj in s2
3.      If (! isSimilar(vi.word, vj.word)
4.        Continue
5.      If (arg_sim(vi, vj, "A0")
6.          && (arg_sim(vi, vj, "A1")
7.          Return co_sim(vi, vj)
8.      Endif
9.    Endfor
10. Endfor
11. Return 0.
```

Algorithm 3: *arg_sim*
Input: Verbs **v1** and **v2** and argument **a**
Output: true if argument **a** is similar in both **v1** and **v2** else false

```
1.  If (v1.arg(a) == null || v2.arg(a) == null)
2.    Return true
3.  Endif
4.  If ( isSimilar(v1.arg(a), v2.arg(a))
5.    Return true
6.  Endif
7.  Return false
```

V. Evaluation Framework

A. Corpus

For evaluation purposes we have used the corpus of plagiarized short answers [13]. It is corpus that has been created by asking five questions to 19 candidates. There are a total of 100 documents with 95 answers provided by the participants and 5 original Wikipedia articles. We have selected the corpus because it provides actual plagiarism cases and has different types of plagiarisms, i.e., simple copy-paste, light revision, heavy revision and no plagiarism. The corpus categorizes the documents into copy-paste plagiarism, where passages were simply copied from the source document without any modifications, light revision plagiarism, where passages were used with slight modifications, heavy revision plagiarism, where the passages were heavily modified and then used, and no plagiarism. Hence it provides the ideal base for testing our model since we aim to target the paraphrasing and synonym usage plagiarism, in this case the light and heavy revision plagiarism.

B. Evaluation Metrics

The standard metrics used for evaluating any plagiarism detection model are recall, precision and f-measure [3]. However for better understanding we have used the positives (where the system detects a case of plagiarism, whether true or not), false positives and false negatives. A case is termed true positive if it is detected to be plagiarized and it is actually a case of plagiarism while as false positive is a case when a sentence is detected to be plagiarized when in reality it is not. True negative is a case when a sentence is deemed non-plagiarized and it is actually non-plagiarized while as it is false negative if it is plagiarized but the system deems it to be non-plagiarized. Using these three metrics we define the recall, precision and f-measure as given below.

$$recall = \frac{\#true\ positives + \#false\ positives}{\#total\ plagiarism\ cases} \quad (2)$$

$$precision = \frac{\#true\ positives}{\#true\ positives + \#false\ positives} \quad (3)$$

$$f - measure = \frac{2 \times recall \times precision}{recall + precision} \quad (4)$$

Recall gives us the fraction of sentences detected to be plagiarized (both correct as well as incorrect) normalized by total number of plagiarized sentences. Precision is the number of correctly detected plagiarized sentences normalized by the total number of plagiarized sentences. F-measure is the harmonic mean of recall and precision.

VI. Results And Analysis

The tests were run as explained in the previous section on the short answers corpus. Since our model is aimed at detecting the intelligent plagiarism, which includes usage of synonyms and/or paraphrasing, we were mainly interested in the document groups labelled as Light Revision and Heavy Revision. Below we describe our findings and the explanations thereof.

Our tests showed near perfect results for two groups of documents. The first group of documents was the "original documents" which had no sign of plagiarism. In this group no false positive was reported because of the stringent condition mentioned in Chapter IV above. Since it is highly unlikely for an original sentence to have verb, agent and patient matching along with other arguments with some source document, it didn't allow false positives. Even if the trio match then the probability that the others arguments match is still minimal. However one case was observed of a false positive and the reason was that the sentence possessed fewer arguments. In fact it had only three arguments. The results are shown in Fig. 2.

The second group was the group of documents labelled "copy-paste" where documents were created by simply copying the passages from some source documents and using them without any change in the plagiarized document. As evident all the arguments would match, hence every case of plagiarism was detected.

The graph in Fig. 3 shows the results obtained for the documents possessing light revision plagiarism. Fig. 11 in appendix II shows the efficiency of detection by showing the number of cases detected by our method, by SRL only method and total cases of plagiarism present. As shown in table V of appendix III the recall is 0.869451 which is remarkably high. The reason is that in light revision the plagiarist actually only uses synonyms and/or paraphrases the sentence, hence the role of various words in the sentence remains the same which is easily detected by our system. There is problem only when the sentence is broken or a sentence is created by combining other sentences. Here the score after passing our criteria goes lower in some cases and hence the reported drop in recall. However it is worthy to note that the results are better than the SRL only method because the plagiarists more often use the pronouns hence difficult for the SRL method to detect.

The group of documents which showed considerable drop in precision values are the heavily revised documents. These documents are more than just synonym usage and paraphrasing. Here sentences are also constructed by the meaning, that is, the meaning is inferred from different passages and then the sentences are written by the plagiarist in his own words. Sometimes the technique of summarization is employed. Due to these techniques the role of words or words themselves does not match. Therefore a considerable drop in precision is reported. One more reason for this relatively low recall is that of changing the noun phrase into verbal phrase and vice versa which results in different set of arguments when SRL is applied. However for documents where these ultra - smart techniques are not used, our system works fairly well. Here also it should be noted that our system works better than the SRL only method. The use of pronouns here is much heavier than the light revision method. Fig. 4 shows the results for this type of plagiarism.

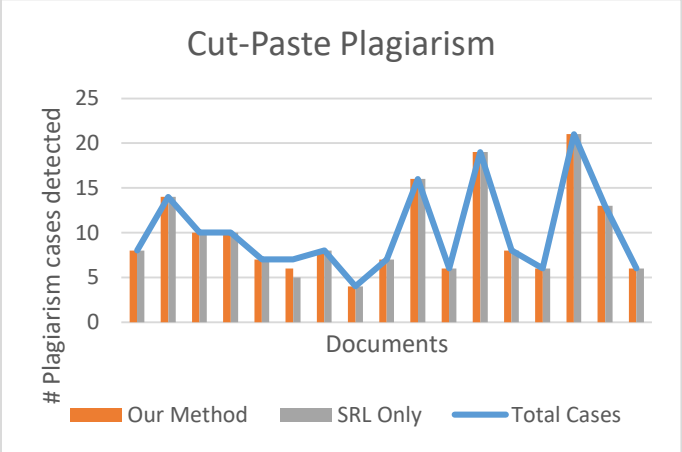


Fig. 2. Gaph showing the results on documents with copy-paste plagiarism employed.

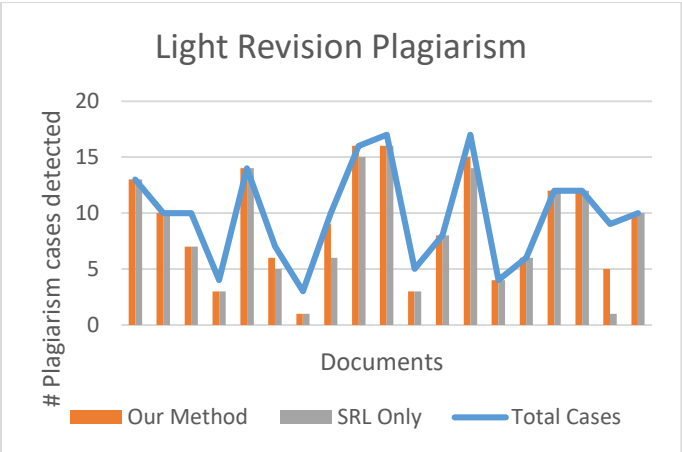


Fig. 3. Gaph showing the results on documents with light revision plagiarism.

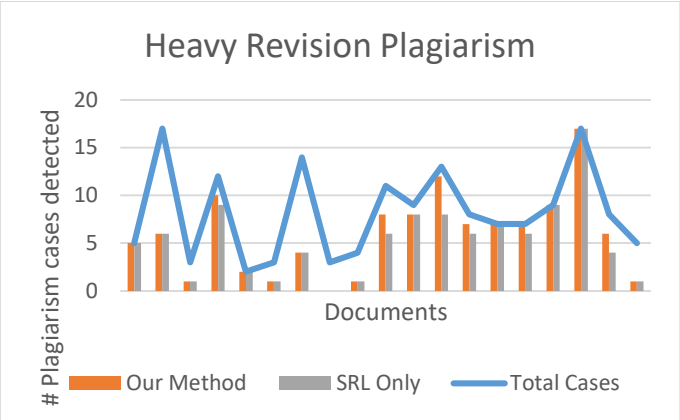


Fig. 4. Gaph showing the results on documents plagiarized with heavy revision.

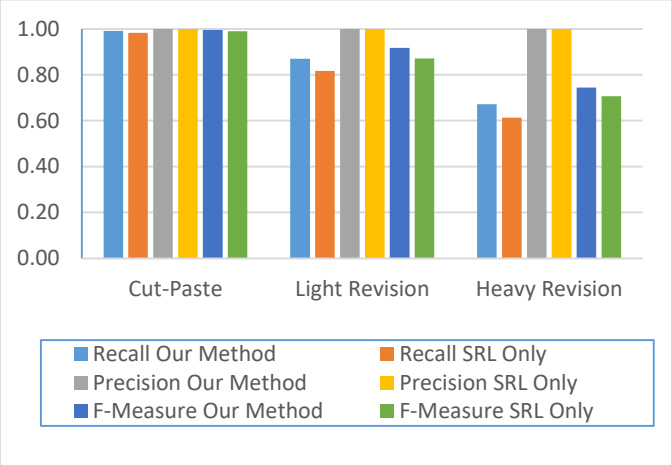


Fig. 5. Gaph showing the recall and f-measure for cut-paste, light revision and heavy revision documents with comparison between our method and SRL only method

TABLE I: COMPARISON BETWEEN OUR PROPOSED METHOD AND OTHER TECHNIQUES USING TIME COMPLEXITY.

Algorithm Time	Complexity
Fuzzy Semantic-based String Similarity[8]	$O(n^2)$
Longest Common Subsequence (LCS) [14]	$O(n^2)$
Semantic-based similarity [9]	$O(n^2)$
SRL-based Similarity (SRL) [3]	$O(n^2)$
Graph-based Method [15]	O(V + E)
Sentence-based Natural Language [16]	$O(n^2)$
Our Method	$O(n^2)$

It should be remarked that our system showed perfect precision value on the test set. However this need not necessarily be true for every case, particularly in documents where there are small sentences with only a few arguments, there are high chances of the system detecting the false positives and hence low precision. The reason is the stringent criteria mentioned in chapter above. Fig. 5 shows the recall, precision and f-measure value for the three groups of documents.

As is the case with many plagiarism detection algorithms our system also falls in the category of those algorithms whose time complexity is $O(n^2)$. Table I shows the complexities of various plagiarism detection algorithms.

VII. CONCLUSION

In this work we have extended the SRL based method for plagiarism detection with three basic improvements. The first basic improvement is that we have developed a basic criteria for similarity based on the fact that object and subject are the most important arguments in a sentence and for two sentences to be similar they must match. The second improvement is in the fact that two sentences to be measured for similarity are often of unequal lengths and we need for similarity measures with normalizing qualities. We have used the cosine similarity for that purpose. Finally, the most important contribution of this work is the use of pronoun resolution which helped in the gain of significant amount of performance over the basic SRL based technique. Although the present work achieves satisfactory results for documents plagiarized with copy-paste technique and with light plagiarisms like synonym usage and paraphrasing, yet for heavy plagiarisms the results are not up to the mark due to the use of verbal phrases instead of verbs.

REFERENCES

[1] L. Gillam and J. Marinuzzi, “Turnitoff – defeating plagiarism detection systems,” in 11th Annual Conference of the Subject Centre for Information and Computer Sciences, Durham, pp. 84–89, 2011.

[2] Alberto Barrón-Cedeño, Parth Gupta, Paolo Rosso, “Methods for cross-language plagiarism detection,” Knowledge-Based Systems, Elsevier, vol 50, pp. 211-217, September 2013.

[3] *A. H. Osman, N. Salim, M. S. Binwahlan, R. Alteeb, and A. Abuobieda, "An improved plagiarism detection scheme based on semantic role labeling," Applied Soft Computing, Elsevier, vol. 12, pp. 1493-1502, 2012.

[4] *Osman, A.H.; Salim, N., "An improved semantic plagiarism detection scheme based on Chi-squared automatic interaction detection," Computing, Electrical and Electronics Engineering (ICCEEE), International Conference on , Khartoum, Sudan, vol., no., pp.640,647, 26-28 Aug. 2013

[5] Alzahrani, S.M.; Salim, N.; Abraham, A., "Understanding Plagiarism Linguistic Patterns, Textual Features, and Detection Methods," Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on , vol.42, no.2, pp.133,149, March 2012

[6] Y. Li, D. Mclean, Z. A. Bandar, J. D. O. Shea, and K. Crockett, “Sentence Similarity Based on Semantic Nets and Corpus Statistics,” Knowledge and Data Engineering, IEEE Transactions on, vol. 18, no. 8, pp. 1138–1150, 2006.

[7] Bao, Jun-Peng, et al. "Semantic sequence kin: A method of document copy detection." Advances in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg, pp. 529-538, 2014.

[8] S. Alzahrani, N. Salim, “Fuzzy Semantic-based String Similarity for Extrinsic Plagiarism Detection,” CLEF, 2010 (Notebook Papers/LABs/Workshops).

[9] K.K. Chow, N. Salim, “Web based cross language plagiarism detection,” Journal of Computing, vol. 1, no. 1, pp. 39-43, December 2009

[10] Mitkov, Ruslan. “Anaphora resolution: the state of the art.” School of Languages and European Studies, University of Wolverhampton, 1999.

[11] P. Elango. "Coreference Resolution: A Survey" in Proc. ACL, Columbus, Ohio, pp. 18-27. 2008.

[12] George A. Miller, “WordNet: A Lexical Database for English,” Communications of the ACM, vol. 38, no. 11: 39-41, 1995

[13] Clough, P. and Stevenson, M., “Developing A Corpus of Plagiarised Short Answers,” Language Resources and Evaluation: Special Issue on Plagiarism and Authorship Analysis, Springer, vol. 45, no. 1, pp. 5-24, 2010.

[14] N.S. Chow Kok Kent, Features based text similarity detection, Journal of Computing, vol. 2, no. 1, pp. 53–57, 2010.

[15] A.H. Osman, N. Salim, S. BinWahlan, H. Hentabli, A.M. Ali, Conceptual similarity and graph-based method for plagiarism detection, Journal of Theoretical and Applied Information Technology, vol. 32, no. 2, pp. 135–145, 2011.

[16] D.R. White, M.S. Joy, Sentence-based natural language plagiarism detection, ACM Journal of Educational Resources, vol. 4, no. 4, December 2004, Pages 1–20.

APPENDIX I

SHOWING EXAMPLES OF VARIOUS METHODS USED IN THE SYSTEM.

Original Sentence:
John is a boy. He loves playing.

After Resolution:
John is a boy. **John** loves playing

Fig. 6. Example of pronoun resolution

The sentence:
The Empress hasn't arrived yet.
After applying SRL:

The Empress	---	**A1 (Agent)**
n't	---	**AM-NEG**
arrive	---	**Verb**
yet	---	**AM-TMP**

	The	Empress	has	n't	arrived	yet
arrive.01	A1			AM-NEG		AM-TMP

Fig. 7. Example of pronoun resolution

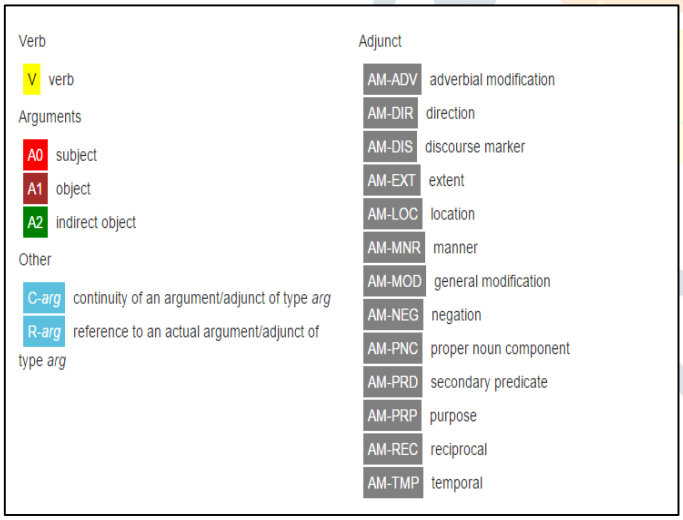


Fig. 9. Meaning of different argument labels taken from SRL demo of University of Illinois[1].

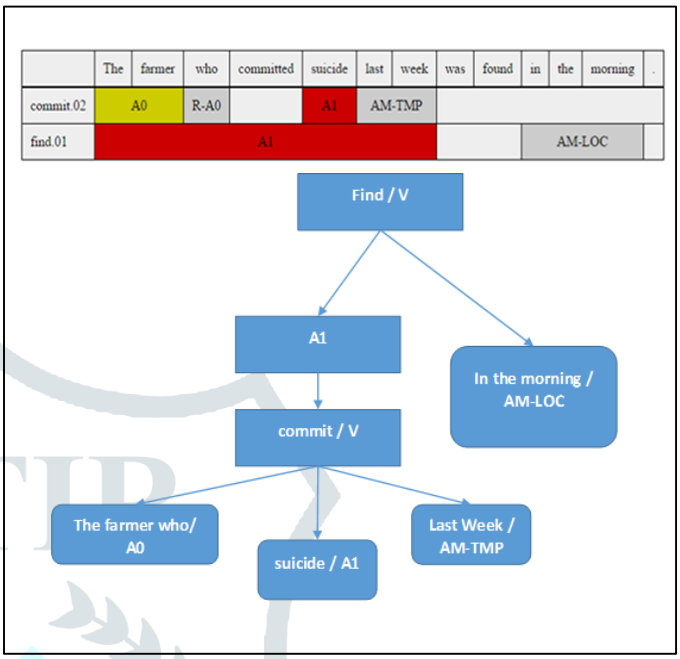


Fig. 8. Construction of verb-tree

[1] http://cogcomp.cs.illinois.edu/page/demo_view/srl

APPENDIX II

GRAPHICAL REPRESENTATION OF THE RESULTS

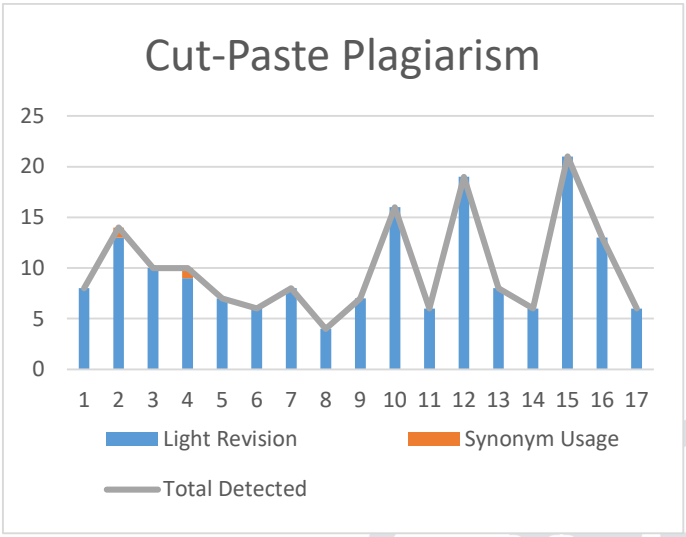


Fig. 10. Gaph showing the results on documents with light revision plagiarism by breaking the total detections between light revision and synonym usage.

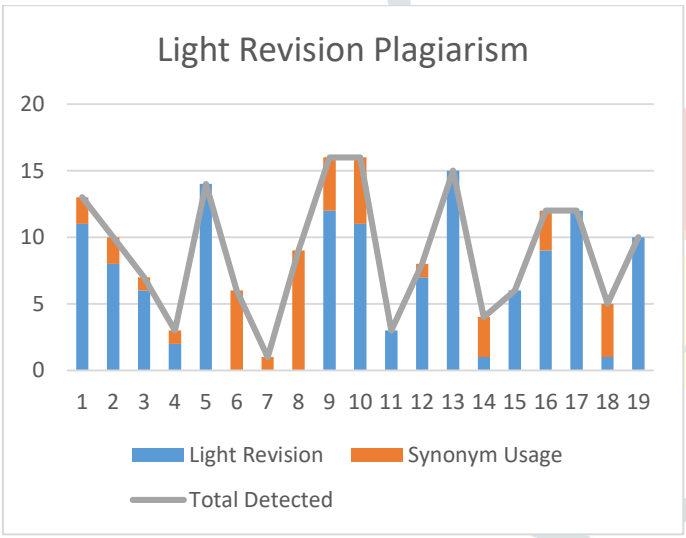


Fig. 11. Gaph showing the results on documents with light revision plagiarism by breaking the total detections between light revision and synonym usage.

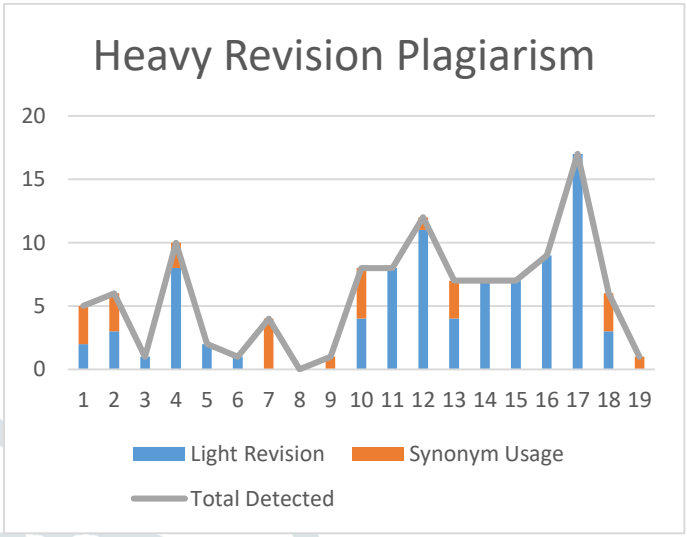


Fig. 12. Gaph showing the results on documents with light revision plagiarism by breaking the total detections between light revision and synonym usage.

APPENDIX III
TABLES SHOWING THE ACTUAL DATA OBTAINED

TABLE II: CUT – PASTE PLAGIARISM RESULTS. PRECISION IS 1 FOR ALL.

		Cut-Paste/ Light Revision		Synonym Usage		Total Detected		Negatives		Recall		F-Measure	
File	Total Cases	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only
g0pA_taskb.txt	8	8	8	0	0	8	8	0	0	1	1	1	1
g0pB_taskc.txt	14	13	13	1	1	14	14	0	0	1	1	1	1
g0pC_taskd.txt	10	10	10	0	0	10	10	0	0	1	1	1	1
g0pD_taska.txt	10	9	9	1	1	10	10	0	0	1	1	1	1
g0pE_taske.txt	7	7	7	0	0	7	7	0	0	1	1	1	1
g1pA_taskd.txt	7	6	5	0	0	6	5	1	2	0.857143	0.714286	0.923077	0.833333
g1pB_taske.txt	8	8	8	0	0	8	8	0	0	1	1	1	1
g1pD_taskb.txt	4	4	4	0	0	4	4	0	0	1	1	1	1
g2pA_taskd.txt	7	7	7	0	0	7	7	0	0	1	1	1	1
g2pB_taske.txt	16	16	16	0	0	16	16	0	0	1	1	1	1
g2pC_taska.txt	6	6	6	0	0	6	6	0	0	1	1	1	1
g3pA_taskd.txt	19	19	19	0	0	19	19	0	0	1	1	1	1
g3pB_taske.txt	8	8	8	0	0	8	8	0	0	1	1	1	1
g3pC_taska.txt	6	6	6	0	0	6	6	0	0	1	1	1	1
g4pB_taske.txt	21	21	21	0	0	21	21	0	0	1	1	1	1
g4pC_taska.txt	13	13	13	0	0	13	13	0	0	1	1	1	1
g4pE_taskc.txt	6	6	6	0	0	6	6	0	0	1	1	1	1
Average										**0.991597**	**0.983193**	**0.995475**	**0.990196**

TABLE III: LIGHT REVISION PLAGIARISM. PRECISION IS 1 FOR ALL.

		Cut-Paste/ Light Revision		Synonym Usage		Total Detected		Negatives		Recall		F-Measure	
File	Total Cases	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only	Our Method	SRL Only
g0pA_taskc.txt	13	11	11	2	2	13	13	0	0	1	1	1	1
g0pB_taskd.txt	10	8	8	2	2	10	10	0	0	1	1	1	1
g0pC_taske.txt	10	6	6	1	1	7	7	3	3	0.7	0.7	0.823529	0.823529
g0pD_taskb.txt	4	2	2	1	1	3	3	1	1	0.75	0.75	0.857143	0.857143
g0pE_taska.txt	14	14	14	0	0	14	14	0	0	1	1	1	1
g1pA_taskc.txt	7	0	0	6	5	6	5	1	2	0.857143	0.714286	0.923077	0.833333
g1pB_taskd.txt	3	0	0	1	1	1	1	2	2	0.333333	0.333333	0.5	0.5
g1pD_taska.txt	10	0	0	9	6	9	6	1	4	0.9	0.6	0.947368	0.75
g2pA_taskc.txt	16	12	11	4	4	16	15	0	1	1	0.9375	1	0.967742
g2pB_taskd.txt	17	11	11	5	5	16	16	1	1	0.941176	0.941176	0.969697	0.969697
g2pC_taske.txt	5	3	3	0	0	3	3	2	2	0.6	0.6	0.75	0.75
g2pE_taskb.txt	8	7	7	1	1	8	8	0	0	1	1	1	1

g3pA_taskc.txt	17	15	14	0	0	15	14	2	3	0.882353	0.823529	0.9375	0.903226
g3pB_taskd.txt	4	1	1	3	3	4	4	0	0	1	1	1	1
g3pC_taske.txt	6	6	6	0	0	6	6	0	0	1	1	1	1
g4pB_taskd.txt	12	9	9	3	3	12	12	0	0	1	1	1	1
g4pC_taske.txt	12	12	12	0	0	12	12	0	0	1	1	1	1
g4pD_taska.txt	9	1	0	4	1	5	1	4	7	0.555556	0.111111	0.714286	0.2
g4pE_taskb.txt	10	10	10	0	0	10	10	0	0	1	1	1	1
Average										**0.869451**	**0.816365**	**0.916979**	**0.871298**

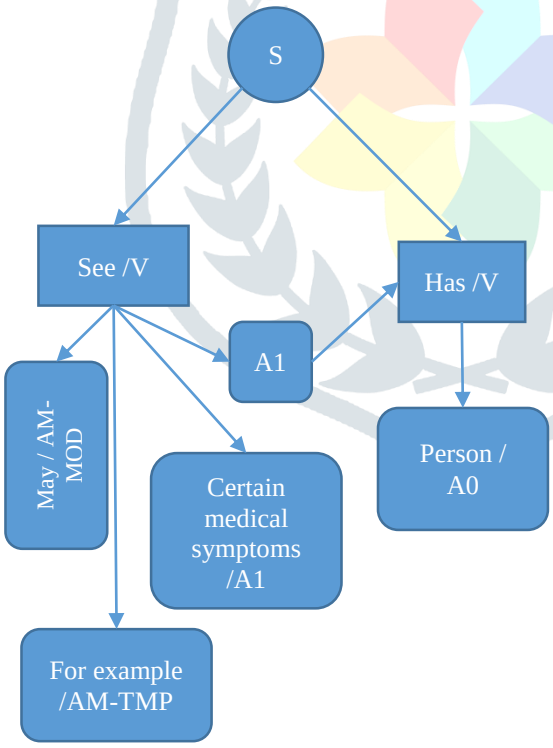
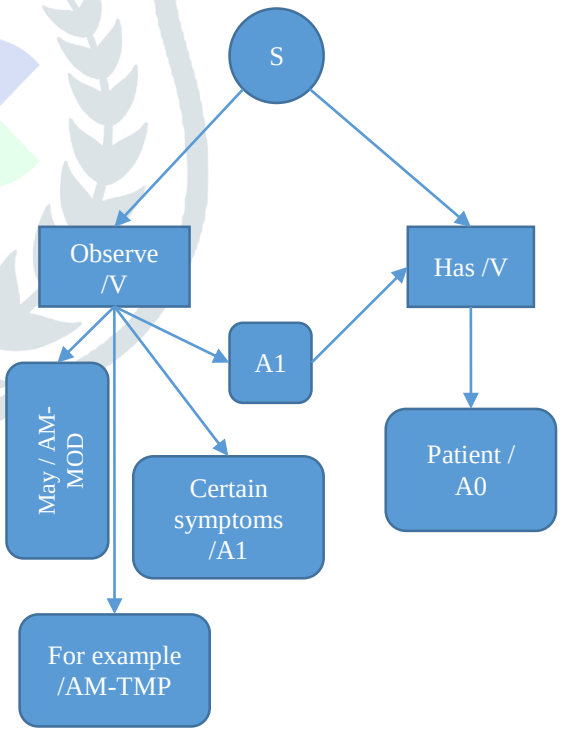
TABLE IV: HEAVY REVISION PLAGIARISM. PRECISION IS 1 FOR ALL.

		Cut-Paste/ Light Revision		**Synonym Usage**		**Total Detected**		**Negatives**		**Recall**		**F-Measure**	
File	**Total Cases**	**Our Method**	**SRL Only**	**Our Method**	**SRL Only**	**Our Method**	**SRL Only**	**Our Method**	**SRL Only**	**Our Method**	**SRL Only**	**Our Method**	**SRL Only**
g0pA_taskd.txt	5	2	2	3	3	5	5	0	0	1	1	1	1
g0pB_taske.txt	17	3	3	3	3	6	6	11	11	0.352941	0.352941	0.521739	0.521739
g0pC_taska.txt	3	1	1	0	0	1	1	2	2	0.333333	0.333333	0.5	0.5
g0pD_taskc.txt	12	8	7	2	2	10	9	2	3	0.833333	0.75	0.909091	0.857143
g0pE_taskb.txt	2	2	2	0	0	2	2	0	0	1	1	1	1
g1pA_taskb.txt	3	1	1	0	0	1	1	3	3	0.333333	0.333333	0.5	0.5
g1pB_taskc.txt	14	0	0	4	4	4	4	10	10	0.285714	0.285714	0.444444	0.444444
g1pD_taske.txt	3	0	0	0	0	0	0	3	3	0	0	0	0
g2pA_taskb.txt	4	0	0	1	1	1	1	3	3	0.25	0.25	0.4	0.4
g2pB_taskc.txt	11	4	3	4	3	8	6	3	5	0.727273	0.545455	0.842105	0.705882
g2pC_taskd.txt	9	8	8	0	0	8	8	1	1	0.888889	0.888889	0.941176	0.941176
g2pE_taska.txt	13	11	7	1	1	12	8	1	5	0.923077	0.615385	0.96	0.761905
g3pA_taskb.txt	8	4	4	3	2	7	6	1	2	0.875	0.75	0.933333	0.857143
g3pB_taskc.txt	7	7	7	0	0	7	7	0	0	1	1	1	1
g3pC_taskd.txt	7	7	6	0	0	7	6	0	1	1	0.857143	1	0.923077
g4pB_taskc.txt	9	9	9	0	0	9	9	0	0	1	1	1	1
g4pC_taskd.txt	17	17	17	0	0	17	17	0	0	1	1	1	1
g4pD_taske.txt	8	3	3	3	1	6	4	2	4	0.75	0.5	0.857143	0.666667
g4pE_taska.txt	5	0	0	1	1	1	1	4	4	0.2	0.2	0.333333	0.333333
Average										**0.671205**	**0.6138**	**0.744335**	**0.705922**

TABLE V: RECALL AND F-MEASURE FOR VARIOUS TYPES OF DOCUMENTS

Type	**Precision**		**Recall**		**F-Measure**	
	Our Method	**SRL Only**	**Our Method**	**SRL Only**	**Our Method**	**SRL Only**
Cut-Paste	1.000000	1.000000	0.991597	0.983193	0.995475	0.990196
Light Revision	1.000000	1.000000	0.869451	0.816365	0.916979	0.871298
Heavy Revision	1.000000	1.000000	0.671205	0.613800	0.744335	0.705922

APPENDIX IV
AN ILLUSTRATION OF HOW THE PROPOSED COMPARISON WORKS.

<table>
<tr><th>S. No.</th><th>Task</th><th>Sentence1</th><th>Sentence2</th></tr>
<tr><td>1</td><td>Sentence source</td><td>Document “g0pB_taskd.txt”</td><td>Original document “orig_taskd.txt”</td></tr>
<tr><td>2</td><td>Sentence</td><td>For example, a person may be seen to have certain medical symptoms</td><td>For example, a patient may be observed to have certain symptoms</td></tr>
<tr><td>3</td><td>SRL</td><td>
<table>
<tr><th>Word</th><th>Label</th><th>Label</th></tr>
<tr><td>For</td><td>AM-DIS</td><td>0</td></tr>
<tr><td>Example</td><td>AM-DIS</td><td>0</td></tr>
<tr><td>A</td><td>A1</td><td>A0</td></tr>
<tr><td>Person</td><td>A1</td><td>A0</td></tr>
<tr><td>May</td><td>AM-MOD</td><td>0</td></tr>
<tr><td>Be</td><td>0</td><td>0</td></tr>
<tr><td>Seen</td><td>V</td><td>0</td></tr>
<tr><td>To</td><td>A1</td><td>0</td></tr>
<tr><td>Have</td><td>A1</td><td>V</td></tr>
<tr><td>Certain</td><td>A1</td><td>A1</td></tr>
<tr><td>Medical</td><td>A1</td><td>A1</td></tr>
<tr><td>symptoms</td><td>A1</td><td>A1</td></tr>
</table>
</td><td>
<table>
<tr><th>Word</th><th>Label</th><th>Label</th></tr>
<tr><td>For</td><td>AM-DIS</td><td>0</td></tr>
<tr><td>Example</td><td>AM-DIS</td><td>0</td></tr>
<tr><td>A</td><td>A1</td><td>A0</td></tr>
<tr><td>Patient</td><td>A1</td><td>A0</td></tr>
<tr><td>May</td><td>AM-MOD</td><td>0</td></tr>
<tr><td>Be</td><td>0</td><td>0</td></tr>
<tr><td>Seen</td><td>V</td><td>0</td></tr>
<tr><td>To</td><td>A1</td><td>0</td></tr>
<tr><td>Have</td><td>A1</td><td>V</td></tr>
<tr><td>Certain</td><td>A1</td><td>A1</td></tr>
<tr><td>symptoms</td><td>A1</td><td>A1</td></tr>
</table>
</td></tr>
<tr><td>4</td><td>Verb-tree structure</td><td> 
</td><td> 
</td></tr>
<tr><td>5</td><td>Matched Verbs</td><td colspan="2">See is included in the synonyms of observe
Sense 19
see -- (observe as if with an eye; "The camera saw the burglary and recorded it")
=> detect, observe, find, discover, notice -- (discover or determine the existence, presence, or fact of; "She detected high levels of lead in her drinking water"; "We found traces of lead in the paint")</td></tr>
</table>

		Has is also matched
6	Argument match for **has**. Therefore printed as a case of plagiarism with score as 1.0.	**Person** is a hypernym of **patient**. 2 senses of patient Sense 1 **patient** -- (a person who requires medical care; "the number of emergency patients has grown rapidly") => case -- (a person requiring professional services; "a typical case was the suburban housewife described by a marriage counselor") => person, individual, someone, somebody, mortal, soul -- (a human being; "there was too much for one person to do")
7	Argument match for **see** and **observe**	Since **see** and **observe** are synonyms, therefore their arguments are also matched. All the arguments match, therefore again a case of plagiarism.