I.J. Intelligent Systems and Applications, **2015, 01, 73-79**
Published Online December 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijisa.2015.01.07

Query Optimization in Arabic Plagiarism Detection: An Empirical Study

Imtiaz H. Khan*, Muazzam A. Siddiqui,**
Kamal M. Jambi*, Muhammad Imran*, Abobakr A. Bagais*
*Department of Computer Science, King Abdulaziz University, Jeddah – KSA
**Department of Information System, King Abdulaziz University, Jeddah – KSA
E-mail: ihkhan@kau.edu.sa; maasiddiqui@kau.edu.sa;
kjambi@kau.edu.sa; imran9pk@gmail.com; abobakr.a.2012@gmail.com

***Abstract* —** This article describes an ongoing research which intends to develop a plagiarism detection system for Arabic documents. We developed different heuristics to generate effective queries for document retrieval from the Web. The performance of those heuristics was empirically evaluated against a sizeable corpus in terms of precision, recall and f-measure. We found that a systematic combination of different heuristics greatly improves the performance of the document retrieval system.

Index Terms—Arabic Plagiarism Detection, Query Generation, Query Optimization, Document Similarity, Arabic Natural Language Processing

I. Introduction

Plagiarism occurs when someone uses the idea or work of another person without proper acknowledgement to the original source. With the advent of the Web, a great magnitude of information is available online thereby increasing the potential sources of plagiarism exponentially. The plagiarism problem has posed a significant threat to academic integrity and the widespread information has made the manual plagiarism detection almost impossible. Automatic plagiarism detection is developed and investigated as a possible countermeasure. In the beginning, the plagiarism detection systems were mostly built to find plagiarism cases in programming languages, for example, Java; later on the research extended to tackle the plagiarism problem in natural languages, for example, English.

We consider the problem of plagiarism detection as falling under the general problem of finding similarity among documents. The problem is well studied in the data/text mining and information retrieval domains [1, 2, 3]. Given a text document, the task of a plagiarism detection system is to find if the document is copied partially or fully from other documents. Broadly, this task is divided into subtasks: extrinsic and intrinsic. The extrinsic (also known as external) plagiarism detection uses different techniques to find similarities among a suspicious document and a reference collection (see, for example, [4, 5, 6]). In this approach each document is represented as an n-dimensional vector where n is the number of terms or some derived features from the document [7]. A number of measures are available to compute the similarity or dissimilarity between vectors including Euclidean distance, Minkowski distance, Mahalanobis distance, Cosine similarity, Simple Matching Coefficient, and Jaccard similarity. On the other hand, in intrinsic plagiarism detection, linguistic features that indicate writing style are used to detect style irregularities in a suspicious document in isolation, without taking a reference collection into account (see, for example, [8, 9]).

We are in the process of developing an online plagiarism detection system for Arabic documents. The proposed plagiarism detection framework comprises of two main components, one *global* and the other *local*. The global component is heuristics-based, in which a given document (henceforth, potential plagiarized document) is used to construct a set of representative queries. These queries are then submitted to Google to retrieve candidate source documents from the Web. Finally, the local component carries out extensive similarity computations to detect if the given document was plagiarized from the documents retrieved from the Web. In this article, we describe our candidate retrieval approach, by presenting a number of heuristics which are developed to construct queries for document retrieval. The effectiveness of this approach is empirically evaluated against a sizeable corpus (details follow in Section IV).

II. Related Work

A large body of work has been published on plagiarism detection [10-27]; for details, see [10]. Here we review a piece of work which is devoted to plagiarism detection in Arabic.

Research in Arabic language reveals that Arabic is a highly inflected language, which constructs its vocabulary through a complicated derivational process using root words. These morphological characteristics and various writing styles pose significant challenges in Arabic language analysis tasks [28-30], including plagiarism detection. For example, the absence of the diacritics could lead to an ambiguous expression, making it extremely difficult to distinguish different words, even in a larger context. The lack of diacritics in most Arabic documents available on the Web is considered as a major challenge to many Arabic NLP tasks.

Very recently, some researchers have shown great interest to develop plagiarism detection systems for Arabic language [31-34]. S. Alzahrani and N. Salim developed an Arabic plagiarism detection system [31] which combines the fuzzy similarity model [35] and semantic similarity model derived from a lexical database [36]. First, they retrieve a list of candidate documents for each suspicious document using shingling and Jaccard coefficient, and then they make sentence-wise detailed comparison between the suspicious and associated candidate documents using the fuzzy similarity model. Their preliminary results indicate that fuzzy semantic-based similarity model can be used to detect plagiarism in Arabic documents. In another study, Bensalem and colleagues [33] have developed a system which uses various stylistic features to account for intrinsic plagiarism. The system was evaluated on a small corpus, so it is difficult to quantify its effectiveness. In yet another study, M. Menai [34] used a top-down approach, whereby in a first step a global similarity is measured between a suspicious document and candidate documents. In a second step, a detailed analysis is done at paragraph- and sentence-level.

Even though we have seen a reasonable work in Arabic plagiarism detection, the research in Arabic plagiarism is still in infancy.

III. Our Candidate Retrieval Approach

A. Measure of Accuracy

Before describing our proposed heuristics for candidate retrieval, we first sketch the concepts we shall be using to measure the performance of each heuristic.We measure the accuracy of a heuristic in terms of precision, recall and f-measure. In our definition of accuracy, a *relevant* URL is the label of the document available in the corpus from which queries are generated. Similarly, a *retrieved* URL is the one within top 10 hits returned by the search engine in response to the submitted query/queries. Accordingly, we define precision and recall as follows.

$$Precision = \frac{|\{Related\ URLs\} \cap \{Retrieved\ URLs\}|}{|\{Retrieved\ URLs\}|}$$

$$Recall = \frac{|\{Related\ URLs\} \cap \{Retrieved\ URLs\}|}{|\{Related\ URLs\}|}$$

We believe that recall is more important to us than precision because if a document is missed at this stage, it cannot be included in the subsequent detailed similarity computation stage, thereby the accuracy of the whole system will be compromised. Accordingly, we use a weighted f-measure statistic with β = 0.25, strongly favoring recall.

$$F - measure = \frac{(1+\beta) * Precision * Recall}{Precision + \beta^2 * Recall}$$

In all our heuristics, we seek to maximize the f-measure: higher the f-measure value, better the heuristic and vice-versa.

B. Optimal query length

When submitting a query to a search engine, it is important to restrict query length to a reasonable number of words. This is because smaller queries result in high recall and low precision while large queries result in low recall and high precision. Larger queries thus seem to be the solution but there are two drawbacks: a) search engines limit the query length to a specified number of characters, and b) very large queries may result in no matching document Therefore, it is imperative to find the query length in a systematic way.

We experimented with different query lengths varying from one word to ten words, where queries were generated by using three different heuristics: first sentence-based heuristic, keyword driven key-phrase based heuristic, and variance in readability scores across sentences (details follow). The combined performance averaged over all three heuristics is shown in Table 1 and Fig. 1. It is clear from Table 1 (and Fig. 1) that the best performance, in terms of f-measure, is achieved by queries of length five or six words. In what follows, we will be using queries of length five (words) only. It is interesting to note that for one-word long queries, recall is very low (just above 3.5%). This is because of our strict definition of retrieved document: we considered only top 10 documents returned by the search engine. For one-word long queries, most queries fail to retrieve the relevant document within top 10 hits.

Table 1. Combined performance of heuristics at different query length (%)

Query length	Precision	Recall	F-Measure
1	3.08	21.86	18.92
2	4.86	72.87	47.02
3	6.18	63.93	48.53
4	14.72	47.38	49.31
5	23.25	46.12	**51.29**
6	35.74	43.28	50.29
7	46.73	39.29	46.66
8	53.92	31.84	38.38
9	61.92	26.26	31.98
10	67.38	17.15	21.10

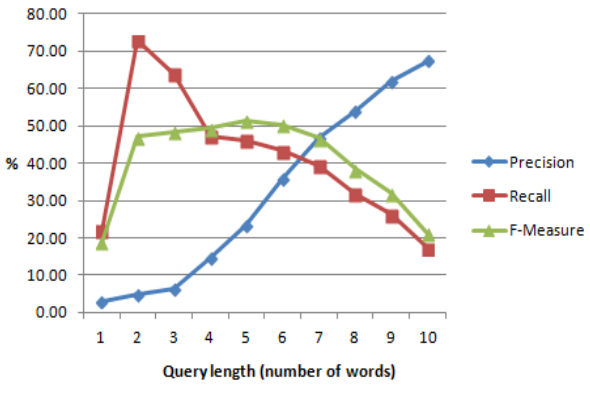

Fig. 1. Combined performance of heuristics at different query length (%)

C. Query generation heuristics

Here we present different heuristics which were considered to construct queries for document retrieval. The performance of each heuristic was empirically evaluated against a sizeable corpus (corpus details follow in Section IV).

1. Variance in readability scores across sentences

Generally, it has been observed that plagiarists copy text from various sources and then edit the copied text. We hypothesize that in such a plagiarized document there would be significant variation in readability scores across different sentences in the document. We also hypothesize that such sentence-level variation can predict potential plagiarism. The main challenge is how to compute the readability score. We explored different readability formulas, including Flesch readability score (Equation 1) and Automated Readability Index (Equation 2).

$$206.835 - 1.015\left(\frac{words}{sentences}\right) - 84.6\left(\frac{syllables}{words}\right) \quad (1)$$

$$4.71\left(\frac{characters}{words}\right) + 0.5\left(\frac{words}{\text{sentences}}\right) - 21.43 \quad (2)$$

Both Flesch readability score and Automated Readability Index are designed to gauge comprehensibility of a text: higher scores indicate that a text is easier to read, whereas lower scores indicate that a text is more difficult to read. Longer sentences yield low score, hence more difficult to comprehend, and vice-versa. In Arabic, it is difficult to identify syllables accurately. Since the Automated Readability Index does not depend on syllables and also it is computationally more efficient, we used it to compute the readability score of a sentence. We used the actual raw text (without any pre-processing) to find the total number of characters, the total number of words and the total number of sentences in a paragraph. Then the readability score of each sentence is computed using the Automated Readability Index. Those sentences whose score falls outside $\mu \pm 2\delta$ (μ: mean, δ: standard deviation) are counted as outliers. These outliers were then used to construct *queries* by taking the first five words of the sentence. The query words were enclosed in double quotes for an exact match (cf. Example 1). The total number of queries for our collection of selected documents varied between 3 to 7 per document; we submitted the first three queries for document retrieval.

Example 1: "تِكْنُولُوجِيا الْمَعْلُومَاتِ فِي حَيَّاتِنَا الْيَوْمِيَّةِ" ("Information technology in our daily lives")

2. Keyword driven key-phrase-based heuristic

We hypothesize that frequency-based keywords in a document can be used to predict the signature of a document to effectively retrieve the potential source documents from the Web. We used a pre-processed document, in which punctuations and stop words were removed, and each word was lemmatized. We sampled a set of top N (N = 5 in this study) distinct keywords, based on the frequency of each word in the entire document. Then, for each keyword we constructed a phrase (henceforth key phrase) by taking two preceding and two succeeding words, at its first appearance in the original document (i.e., without preprocessing). If the keyword appeared at the beginning (or end) of a sentence, four preceding (or four succeeding words) words were used to construct the key phrase. This heuristic yielded five queries per document. Again, the query words were enclosed in double quotes for an exact matching (cf. Example 2, in which the keyword is underlined).

Example 2: "جيل من الطلاب قادر على" ("A generation of students capable of")

3. First-sentence-based heuristic

We conjecture that the likelihood of the first sentence being plagiarized would be more as compared to the other sentences in the same paragraph. Therefore, we constructed a query by taking the first five words of the first sentence of each paragraph (cf. Example 3). The total number of queries for our collection of selected documents varied between 2 to 5 per document; we submitted all the queries generated for a given suspicious document for source document retrieval.

Example 3: "خصائص مشتركة بين معالجات النصوص" ("Common features between text processing tools")

4. Random-sentence-based heuristic

In this heuristic, N (N = 5, in this study) sentences were selected randomly from each document. Then, a query was constructed for each sentence by taking the first five words of the sentence, where the query words were enclosed in double quotes (cf. Example 4). In this study, the random-sentence based heuristic is taken as baseline.

Example 4: "الذي يساعد في اتخاذ القرارات" ("That helps in making decision")

IV. Empirical Study

A. Corpus

We developed a corpus consisting of assignments submitted by students in an introduction to information technology course. The corpus statistics are displayed in Table 2. (The detailed methodology of corpus construction is reported elsewhere in [37].) The students were encouraged to use the Web and provide the URLs of the web pages consulted in solving the assignment. These URLs serve two purposes, one as a label indicating that the document is plagiarized from the Web, and two, to download the source document (webpage) from the Web, if possible, for further analysis.

B. Methodology

We developed an information retrieval system as shown in Fig. 2. The system takes a suspicious document d as an input and goes through the following steps to find potential source documents.

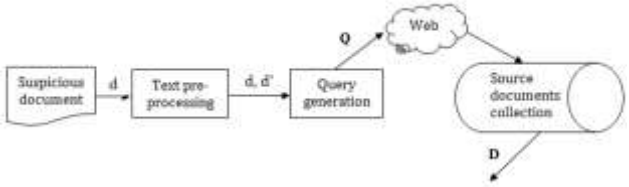

Fig. 2. Information retieval system to search for source documents on the Web for a given suspicious document

Table 2. Corpus statistics

Type	Count	Proportion
Total documents in the corpus	1156	
Plagiarized documents	892	77.2% of total
Not plagiarized documents	264	22.8% of total
Documents plagiarized from the Web	718	80.5% of plagiarized
Documents plagiarized from other sources	174	19.5% of plagiarized
Documents plagiarized from the Web with source URL provided	551	76.7% of web plagiarized
Documents plagiarized from the Web without source URL provided	167	23.3% of web plagiarized

1. We start by pre-processing a suspicious document d. The original document d is tokenized and the stopwords are removed to generate d'.
2. Different query generation heuristics are used to generate a set of queries Q from the given suspicious document d. The query generation module takes the document d, the pre-processed document d' and the query generation heuristic as input and returns a set of queries Q as output. In this study, we used 35 documents plagiarized from the Web; the documents were selected pseudorandomly from our own corpus. The same 35 documents were used to generate queries by each heuristic.
3. Google's custom search API was used to search Q on the Web.

It is instructive to describe how the Google search API is used to retrieve the source documents from the Web. The Google search API needs API key, a search engine ID and a response format. We used a vendor provided API key and search engine ID, and the response format was set to JSON. With these credentials, each query (from the set Q) was then submitted to the Web in turn. For each query, the top 10 hits were saved for further analysis. In case there were fewer than 10 hits, all the returned documents were saved. Finally, the URL of each saved document, i.e. the relevant URL, was extracted for further analysis. We repeated this process for all queries generated by each heuristic in turn.

C. Results and Analysis

For each query, we recorded the relevant URL of the suspicious document (provided as a label in the corpus) from which the query was generated, the heuristic itself, and the retrieved URL (returned as a search result). The data were recorded for a set of 35 plagiarized documents. We report on precision, recall and f-measure of each heuristic. The results are shown in Table 3 and Fig. 3.

Table 3. Performance of heuristics (%)

Heuristic	Precision	Recall	F-measure	F-measure above baseline
Random sentence (R)	16.36	14.75	17.46	
First sentence (F)	18.46	19.35	22.71	5.25
Key phrases (K)	33.85	36.67	42.93	25.47
Variance in RS (V)	26.57	22.17	24.7	7.29
F + K	38.57	45.00	52.43	34.97
F + V	27.14	31.67	36.89	19.43
K + V	34.29	40.00	46.60	29.14
F + K + V	36.25	48.33	55.77	**38.31**

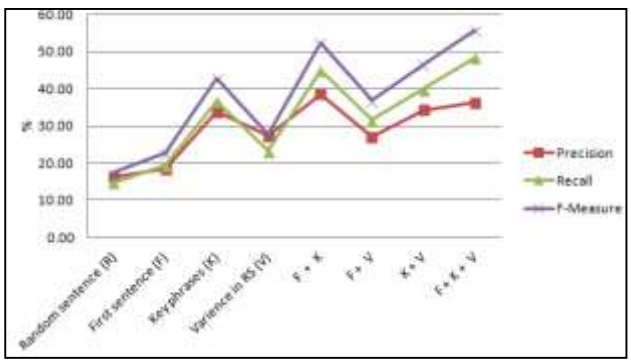

Fig. 3. Performance of heuristics (%)

We also combined the results of the individual heuristics, excluding the random-sentence based heuristic, which we considered as baseline. It is clear that a combination of these heuristics improves the f-measure (Table 3 and Fig. 3). For example, a combination of all three heuristics (F + K + V) gave much better results; in this case f-score is 38.31% above the baseline. This suggests that the different heuristics have different predictive power and a systematic combination of these heuristics can greatly improve the overall performance of the document retrieval component. It is also interesting to note that the results of the key-phrase based heuristic are heuristics are far better than the baseline results (25.47% above the baseline). However, the individual performance of the first-sentence based heuristic and the variance in readability score based heuristic is only marginally better than the baseline (5.25% and 7.29% above the baseline, respectively).

V. General Discussion

Automatic plagiarism detection is a well studied problem, however, the challenge still remains how to search the potential source documents in the first place from the Web before applying the detailed similarity analysis. Over the past one decade or so, researchers have focused on improving the quality of the Web search. This is important because different search engines not only place restrictions on the number of queries submitted to the Web per day, the search engines have to respond in interactive response time. Query optimization has been suggested as a countermeasure to combat these issues. In this study, we proposed different heuristics to generate effective queries for document retrieval. The results show that performance of each heuristic is above the baseline, the key-phrase based heuristic offering the best performance. The results also indicate that a combination of different heuristics greatly improves the performance of the document retrieval system. This study raised some interesting questions.

1. Different search APIs are available but they have different limitations, including restrictions on the maximum number of queries per day, limited search results, and mostly confined to English language only. We explored the possibility of using Duck Duck Go [38], Faroo [39] and Blekko [40], and found that the Google's custom search API is the most suitable for our task. The Google search API allows a maximum 100 queries per day for free, however, with a reasonable subscription this restriction is waived off.
2. The documentation of Google search API reveals that the maximum allowed query length is 2048 characters. However, we observed that for Arabic, even for a 12-words long query, the API throws a 'query too long' error. This is an important limitation.
3. The Flesch readability formula uses syllables in its computation. However, we were unable to find a tool which could identify syllables accurately in Arabic.

VI. Conclusion and Future Work

In this study, we developed different heuristics to generate effective queries for document retrieval. These heuristics include key-phrases informed by frequency-based keywords, variance in readability score and first sentence in a paragraph. The performance of these heuristics was empirically evaluated by using our own corpus in terms of precision, recall and f-measure. We found that individually the key-phrases based heuristic offers better performance, however, the overall performance of the document retrieval system greatly improves by combining the different heuristics.

The work presented here is part of plagiarism detection system for Arabic documents. The present work only discusses the candidate document retrieval system. We intend to integrate this with a detailed similarity computation system to detect if a given document was plagiarized from the candidate documents retrieved from the Web. This would shed more light on the overall performance of our heuristic-based document retrieval approach for plagiarism detection.

Acknowledgments

This work was supported by a King Abdulaziz City of Science and Technology (KACST) funding (Grant No. 11-INF-1520-03). We thank KACST for their financial support.

References

[1] M. Konchady. Buiding search applications: Lucene, LingPipe, and Gate. First Edition. Mustru Publishing, 2008.
[2] P.-N. Tan, M. Steinbach, and V. Kumar. Introduction to data mining. Addison Wesley, 2005.
[3] M. D. Lee, B. Pincombe, and M. Welsh. An empirical evaluation of models of text document similarity. In B. G. Bara, L. Barsalou, and M. Bucciarelli, editors, 27th Annual Meeting of the Cognitive Science Society, pages 1254–1259. 2005.
[4] C.-H. Leung and Y.-Y. Chan. A natural language processing approach to automatic plagiarism detection. In Proceedings of the 8th ACM SIGITE conference on information technology education, pages 213–218, 2007.
[5] T. Wang, X. Z. Fan, and J. Liu. Plagiarism detection in chinese based on chunk and paragraph weight. In Proceedings of the 7th International Conference on Machine Learning Cybernet, pages 2574–2579, Beijing, China, 2008.
[6] J. A. Malcolm and P. C. R. Lane. Tackling the pan09 external plagiarism detection corpus with a desktop plagiarism detector. In Proceedings of SEPLN, pages 29–33, Spain, 2009.
[7] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. Communications of the ACM, 18(11): 613–620, 1975.
[8] M. Eissen, B. Stein, and M. Kulig. Plagiarism detection without reference collections. In Proceedings of the advances in data analysis, pages 359–366, 2007.
[9] S. Benno, K. Moshe, and S. Efstathios. Plagiarism analysis, authorship identification, and near-duplicate detection. In Proceedings of the ACM SIGIR Forum PAN07, pages 68–71, New York, 2007.
[10] S. M. Alzahrani, N. Salim, and A. Abraham. Understanding plagiarism linguistic patterns, textual features and detection methods. 42, 2012.
[11] S. Brin, J. Davis, and H. Garcia-Molina. Copy detection mechanisms for digital documents. In Proceedings of the ACM SIGMOD annual conference., 1995.
[12] A. Z. Broder. On the resemblance and containment of documents. Compression and complexity of sequences, pages 21–29, 1997.
[13] J. Kasprzak, M. Brandejs, and M. Kˇripaˇc. Finding plagiarism by evaluating document similarities. In Proceedings of the SEPLN, pages 24–28, Spain, 2009.
[14] A. Si, H. V. Leong, and R. W. Lau. check: A document plagiarism detection system. In Proceedings of ACM symposium for applied computing, pages 70–77, 1997.
[15] D. Zou, W. Long, and Z. Ling. A cluster-based plagiarism detection method. In Proceedings of the CLEF Workshop, 2010.
[16] M. Elhadi and A. Al-Tobi. Use of text syntactical structures in detection of document duplicates. In

Proceedings of the third international conference on digital information management, pages 520–525, 2008.
[17] S. Torres and A. Gelbukh. Comparing similarity measures for original WSD lesk algorithm. In Proceedings of advances in computer science and application, 2009.
[18] N. Shivakumar and H. Garcia-Molina. Building a scalable and accurate copy detection mechanism. In Proceedings of the 1st ACM international conference on digital libraries, 1996.
[19] K. Monostori, A. Zaslavsky, and H. Schmidt. A repetition based measure for verification of text collections and for text categorization. In Proceedings of information resources management association international conference, pages 955–957, 2000.
[20] D. Khmelev and W. Teahan. A repetition based measure for verification of text collections and for text categorization. In Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval, pages 104–110, 2003.
[21] R. L. Ribler and M. Abrams. Using visualization to detect plagiarism in computer science classes. In Proceedings of IEEE symposium on information visualization, pages 173–178, 2000.
[22] R. Lukashenko, V. Graudina, and J. Grundspenkis. Computer-based plagiarism detection methods and tools: an overview. In Proceedings of the ACM international conference on computer systems and technologies, pages 203–215, 2007.
[23] P. Clough. Old and new challenges in automatic plagiarism detection. Technical report, national plagiarism advisory service, 2003.
[24] P. Runeson, M. Alexanderson, and O. Nyholm. Detection of duplicate defect reports using natural language processing. In Proceedings of the 29th international conference on software engineering, pages 499–510, 2007.
[25] I. Androutsopoulos and P. Malakasiotis. A survey of paraphrasing and textual entailment methods. Technical report, Athens university of economics and business, Greece, 2009.
[26] Z. Ceska and C. Fox. The influence of text pre-processing on plagiarism detection. In Proceedings of the recent advances in natural language processing, 2009.
[27] M. Chong, L. Specia, and R. Mitkov. Using natural language processing for automatic detection of plagiarism. In Proceedings of the 4th international plagiarism conference, 2010.
[28] A. N. De Roeck and W Al-Fares. A morphologically sensitive clustering algorithm for identifying Arabic roots. In Proceedings of the association for computational linguistics (ACL'00), 2000.
[29] Rozovskaya, R. Sproat, and E. Benmamoun. Challenges in processing colloquial Arabic: The challenge of Arabic for NLP/MT. In international conference at the British computer society, London, UK, 2006.
[30] N. Habash. Arabic tutorial. In the fifth international conference on Language Resources and Evaluation, LREC'06, 2006.
[31] S. Alzahrani and N. Salim. Fuzzy semantic-based string similarity for extrinsic plagiarism detection. In Proceedings of the 2nd international conference on the applications of digital information and Web technologies., London, UK, 2009.
[32] A. Jadalla and A. Elnagar. A fingerprinting-based plagiarism detection system for Arabic text-based documents. In Proceedings of the 8th international conference on computing technology and information management, 2012.
[33] I. Bensalem, P. Rosso, and S. Chikhi. Intrinsic plagiarism detection in Arabic text: Preliminary experiments. In Proceedings of the 2nd Spanish conference on information retrieval, Spain, 2012.
[34] M. Menai. Detection of plagiarism in Arabic documents. International journal of information technology and computer science (IJITCS), 4(10), 2012.
[35] R. Yerra and Y. Ng. A sentence-based copy detection approach for web documents. Fuzzy systems and knowledge discovery, pages 557–570, 2005.
[36] Y. Li, D. McLean, Z. A. Banda, J. D. O'Shea, and K. Crockett. Sentence similarity based on semantic nets and corpus statistics. IEEE transactions on knowledge and data engineering, 18(8):1138–1150, 2006.
[37] M. A. Siddiqui, S. Elhag, I. H. Khan and K. Jambi. Building an Arabic plagiarism detection corpus. To appear in language resources and engineering.
[38] Duck Duck Go. [Online]. Available: https://duckduckgo.com/api. [Accessed: 13-Jan-2014].
[39] Faroo Web Search. [Online]. Available: http://www.faroo.com/hp/api/api.html. [Accessed: 13-Jan-2014].
[40] Blekko API. [Online]. Available: http://help.blekko.com/index.php/does-blekko-have-an-api/. [Accessed: 01-Dec-2013]

Authors' Profiles

Imtiaz Hussain Khan is an assistant professor in Department of Computer Science at King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia. He received his MS in Computer Science from the University of Essex UK in 2005 and PhD in Natural Language Generation from the University of Aberdeen UK in 2010. His areas of research are Natural Language Processing and Evolutionary Computation.

Muazzam Ahmed Siddiqui is an assistant professor at the Faculty of Computing and Information Technology, King Abdulaziz University. He received his BE in electrical engineering from NED University of Engineering and Technology, Pakistan, and MS in computer science and PhD in modeling and simulation from University of Central Florida. His research interests include text mining, information extraction, data mining and machine learning.

Prof. Kamal M. Jambi is a professor in Department of Computer Science at King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia. He received his Masters (Computer Science) degree from Michigan State University, USA in 1986. He earned his PhD (Artificial Intelligence, OCR) degree from the Illinois Institute of Technology, Chicago, USA in 1991. His areas of research are Natural Language Processing, Speech Recognition , OCR and image processing.

Muhammad Imran is a research assistant in Department of Computer Science at King Abdulaziz University, Jeddah, Kingdom of Saudi Arabia. He received his Bachelors (Software Engineering) degree from International Islamic University, Pakistan in 2010 and his Masters (Computer Science) degree from the King Abdulaziz University, KSA in 2014. His areas of research include Software Engineering and Natural Language Processing.

Abobakr A. Bagais received his BSc degree in Computer Science from King Abdul-Aziz University, Saudi Arabia. He is currently pursuing M.E. in Network optimization from King Abdul-Aziz University. His areas of interest include Arabic natural language processing, bioinformatics and optimization network.

How to cite this paper: Imtiaz H. Khan, Muazzam A. Siddiqui, Kamal M. Jambi, Muhammad Imran, Abobakr A. Bagais,"Query Optimization in Arabic Plagiarism Detection: An Empirical Study", International Journal of Intelligent Systems and Applications (IJISA), vol.7, no.1, pp.73-79, 2015. DOI: 10.5815/ijisa.2015.01.07

