ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

An Analysis on Indexing Techniques for Scalable Record Linkage, Data Leakage and De-duplication in World Wide Web

M. Prem Kumar, H. Vignesh Ramamoorthy

Head, Dept. of Computer Technology, Sree Saraswathi Thyagaraja College, Pollachi, India

Assistant Professor, Dept. of Computer Science, SreeSaraswathiThyagaraja College, Pollachi, India

ABSTRACT: The primary intent of this research is to develop a novel and efficient approach for detection of near duplicates in web documents. Initially the crawled web pages are pre-processed using document parsing which removes the HTML tags and java scripts present in the web documents. This is followed by the removal of common words or stop words from the crawled pages. Then the stemming algorithm is applied to filter affixes (prefixes and the suffixes) of the crawled documents in order to get the keywords. Finally, the similarity score between two documents is calculated on basis of the extracted keywords. The documents with similarity scores greater than a predefined threshold value are considered as near duplicates, in this research we have conducted an extensive experimental study using several real datasets, and have demonstrated that the proposed algorithms outperform previous ones.The voluminous amounts of web documents swarming the web have posed huge challenges to the web search engines making their results less relevant to the users. The presence of duplicate and near duplicate web documents in abundance has created additional overheads for the search engines critically affecting their performance and quality. The detection of duplicate and near duplicate web pages has long been recognized in web crawling research community. It is an important requirement for search engines to provide users with the relevant results for their queries in the first page without duplicate and redundant results.

KEYWORDS: Duplicates, HTML tags, Java Scripts, Stemming, Threshold.

I. INTRODUCTION

The quick expansion of information sources present on the World Wide Web has necessitated the users to make use of automated tools to locate desired information resources and to follow and assess their usage patterns. Thus the need for building server side and client side intelligent systems can mine for knowledge in a successful manner. A portion of data mining that revolves around the assessment of World Wide Web is known as Web mining. Data Mining, Internet technology, World Wide Web as well as Semantic Web, are incorporated in Web mining. Web mining refers to "the use of data mining techniques to automatically discover and extract information from World Wide Web documents and services". Web usage mining, web structure mining and web content mining are the three areas into which web mining has been classified into. The process of information detection from millions of sources across the World Wide Web is known as Web content mining[1]. The past few years have observed the drastic development of the World Wide Web (WWW). Information is being increasingly accessible on the web. The performance and scalability of the web engines face considerable problems due to the presence of enormous amount of web data[2][3].The expansion of internet has resulted in problems for the Search engine owing to the fact that the flooded search results are of less relevance to the users. Any one of the subsequent features: different character sets, formats, and inclusions of advertisement or current date may be the reason behind the dissimilarity among identical pages served from the same server[2]. Web crawling is employed by the search engines to populate a local indexed repository of web pages which is in turn utilized to answer user search queries.

Business has become more proficient and fruitful owing to the ability to access contents of interest amidst huge heaps of data. Web crawling forms an integral component for search engines. A program or automated script that traverses the World Wide Web in a systematic, automated manner is known as a web crawler or web spider or web robot. Web crawlers are also known by other names like ants, automatic indexers, bots, and worms. Web crawlers aid

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

in the creation of web pages that proffer input for systems that index, mine or else analyze pages (e.g. a web search engine). Documents and links related to diverse topics are crawled by the Generic crawlers while precise knowledge is use to restrict the focused crawlers to crawl only specific topics. Issues such as freshness and efficient resource usage have previously been overcome. On the other hand the issue of near duplicate web document elimination in generic crawl still remains unaddressed. Web contains duplicate pages and mirrored web pages in abundance. Standard check summing techniques can facilitate the easy recognition of documents that are duplicates of each other (as a result of mirroring and plagiarism). The efficient identification of near duplicates is a vital issue that has arose from the escalating amount of data and the necessity to integrate data from diverse sources and needs to be addressed[3]. Though near duplicate documents display striking similarities, they are not bit wise similar.

Web search engines face considerable problems due to duplicate and near duplicate web pages. These pages enlarge the space required to store the index, either decelerate or amplify the cost of serving results and so exasperate users. Thus algorithms for recognition of these pages become inevitable. Due to high rate of duplication in Web document the need for detection of duplicated and nearly duplicated documents is high in diverse applications like crawling, ranking clustering archiving and caching[3][4]. Nevertheless the performance and scalability of the duplicate document detection algorithms is affected by the huge number of web pages. Search engines like Google encounter numerous duplicate and near duplicate pages while crawling the Web yet they inhibit the return of such pages in search results so as to provide the users with distinct and beneficial information on the first page. Despite the fact that near duplicates are not bit wise identical, they are strikingly similar[4]. Near duplicates possess minute difference and so are not regarded are exact duplicates. Typographical errors, versioned, mirrored, or plagiarized documents, multiple representations of the same physical object, spam emails generated from the same template and the like are some of the chief causes for the prevalence of near duplicate pages. Such near duplicates contain similar content and vary only in minimal areas of the document like the advertisements, counters and timestamps. Web searches consider these differences as inappropriate. Various studies have identified a substantial portion of web pages as near duplicates. According to these studies the web pages traversed by crawlers comprises of 1.7% to 7% of near duplicates. Conservation of network bandwidth, reduction in storage costs and enhancement in the standard of search indexes can be achieved with the elimination of near duplicates. Besides, the load on the remote host that serves the web pages is also decreased. Near duplicate page detection systems are prone to numerous challenges. First is the concern of scale: search engines index billions of web-pages; this results in a multi-terabyte database. Second issue is the ability of the crawl engine to crawl billions of web-pages every day[4].

II. RELATED WORK

The creation of a consensus corpus that was obtained through combining three individual annotations of the same clinical corpus in Swedish. We used a few basic rules that were executed automatically to create the consensus. The corpus contains negation words, speculative words, uncertain expressions and certain expressions[5]. We evaluated the consensus using it for negation and speculation cue detection[6]. We used Stanford NER, which is based on the machine learning algorithm Conditional Random Fields for the training and detection. For comparison we also used the clinical part of the Bioscope Corpus and trained it with Stanford NER. For our clinical consensus corpus in Swedish we obtained a precision of 87.9 percent and a recall of 91.7 percent for negation cues, and for English with the Bioscope Corpus we obtained a precision of 97.6 percent and a recall of 96.7 percent for negation cues[7].This tutorial describes what Denial of Service (DOS) attacks are. How they can be carried out in IP networks, and how one can defend against them. Distributed DoS (DDoS) attacks are included here as a subset of DoS attacks. A DoS attack has two phases: a deployment and an attack phase. A DoS program must first be deployed on one or more compromised hosts before an attack is possible[8]. Mitigation of DoS attacks requires thus defense mechanisms for both phases. Completely reliable protection against DoS attacks is, however, not possible. There will always be vulnerable hosts in the Internet. And many attack mechanisms are based on ordinary use of protocols, Defense in depth is thus needed to mitigate the effect of DoS attacks. This paper describes shortly many defense mechanisms proposed in the literature.

The goal is not to implement all possible defenses. Instead, one should optimize the trade-off between security costs and acquired benefits in handling the most important risks. Mitigation of DoS attacks is thus closely related to risk management.Intrusion detection faces a number of challenges; an intrusion detection system must reliably detect malicious activities in a network and must perform efficiently to cope with the large amount of network traffic. In this paper, we address these two issues of Accuracy and Efficiency using Conditional Random Fields and Layered

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

Approach[9]. We demonstrate that high attack detection accuracy can be achieved by using Conditional Random Fields and high efficiency by implementing the Layered Approach. Experimental results on the benchmark KDD '99 intrusion data set show that our proposed system based on Layered Conditional Random Fields outperforms other well-known methods such as the decision trees and the naive Bayes. The improvement in attack detection accuracy is very high, particularly, for the U2R attacks (34.8 percent improvement) and the R2L attacks (34.5 percent improvement). Statistical Tests also demonstrate higher confidence in detection accuracy for our method. Finally, we show that our system is robust and is able to handle noisy data without compromising performance[10].The authors' perspective of database mining as the confluence of machine learning techniques and the performance emphasis of database technology is presented. Three classes of database mining problems involving classification, associations, and sequences are described. It is argued that these problems can be uniformly viewed as requiring discovery of rules embedded in massive amounts of data. A model and some basic operations for the process of rule discovery are described[9][10]. It is shown how the database mining problems considered map to this model, and how they can be solved by using the basic operations proposed. An example is given of an algorithm for classification obtained by combining the basic rule discovery operations. This algorithm is efficient in discovering classification rules and has accuracy comparable to ID3, one of the best current classifiers[11].Although an intelligent intrusion and detection strategies are used to detect any false alarms within network critical segments of network infrastructures, reducing false positives are still being a major challenges. Up to this moment, these strategies focus on either detection or response features, but often lack of having both features together. Without considering those features together, intrusion detection systems are probably cannot highly detect on low false alarm rates[12]. To offset abovementioned constraints, this paper proposes a strategy to focus on detection involving statistical analysis of both attack and normal traffics based on the training data of KDD Cup 99. This strategy is also included a hybrid statistical approach which using Data Mining and Decision Tree Classification[13][14]. As a result, the statistical analysis can be manipulated to reduce misclassification of false positives and distinguish between attacks and false positives for the data of KDD Cup 99. Therefore, this strategy can be used to evaluate and enhance the capability of the IDS to detect and at the same time to respond to the threats and benign traffic in critical segments of network, application and database infrastructures[15].

III. PROPOSED WORK

A. Web Mining;

Web mining is the application of data mining techniques to discover patterns from the Web. Web mining can be divided into three different types, which are Web usage mining, Web content mining and Web structure mining. Web mining is a technique to search, collate and analyze patterns in the data content of web sites by using traditional data mining techniques and attributes such as clustering and classification, association, and examination of sequential patterns[16]. Web Mining tools analyze web logs for useful customer-related information that can help personalize web sites according to user behavior. Web mining tools are also used to search the web for key words, phrases, or other content.

B. Search Engine

A web search engine is a tool designed to search for information on the World Wide Web. The search results are usually presented in a list of results and are commonly called hits. The information may consist of web pages, images, information and other types of files. Some search engines also mine data available in databases or open directories. Unlike Web directories, which are maintained by human editors, search engines operate algorithm epically or are a mixture of algorithmic and human input[16]. Web search engines work by storing information about many web pages, which they retrieve from the html itself. These pages are retrieved by a Web crawler (sometimes also known as a spider) — an automated Web browser which follows every link on the site. Exclusions can be made by the use of robots.txt[17]. The contents of each page are then analysed to determine how it should be indexed (for example, words are extracted from the titles, headings, or special fields called Meta tags). Data about web pages are stored in an index database for use in later queries. A query can be a single word. The purpose of an index is to allow information to be found as quickly as possible.

C. Near Duplicate Detection

We possess the distinct keywords and their counts in each of the each crawled web page as a result of stemming. These keywords are then represented in a form to ease the process of near duplicates detection. This representation will reduce the search space for the near duplicate detection[18]. Initially the keywords and their number of occurrences in a

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

web page have been sorted in descending order based on their counts. Afterwards, n numbers of keywords with highest counts are stored in a table and the remaining keywords are indexed and stored in another table. In our approach the value of n is set to be 4. The similarity score between two documents can be calculated if and only the prime keywords of the two documents are similar. Thus the search space is reduced for near duplicates detection.

D. Search Result

If the prime keywords of new page are same with a page in the repository, then the similarity score between the two documents is calculated. The keywords in the tables are considered individually for the similarity score calculation. If a keyword is present in both the tables, the formula used to calculate the similarity[19]. The web documents with similarity score greater than a predefined threshold are near duplicates of documents already present in repository. These near duplicates are not added in to the repository for further process such as search engine indexing.

E. Process Description

The near duplicate detection is performed on the keywords extracted from the web documents. First, the crawled web documents are parsed to extract the distinct keywords. Parsing includes removal of HTML tags, java scripts, stop words/common words and stemming of remaining words. The extracted keywords and their counts are stored in a table to ease the process of near duplicates detection. The keywords are stored in the table in a way that the search space is reduced for the detection[20]. The similarity score of the current web document against a document in the repository is calculated from the keywords of the pages. The documents with similarity score greater than a predefined threshold are considered as near duplicates.

F. Near Duplicate Web Documents

Even though the near duplicate documents are not bitwise identical they bear striking similarities. The near duplicates are not considered as "exact duplicates" but are files with minute differences. Typographical errors, versioned, mirrored, or plagiarized documents, multiple representations of the same physical object, spam emails generated from the same template, and many such phenomenon's may result in near duplicate data. These studies propose that near duplicates constitute almost 1.7% to 7% of the web pages traversed by crawlers. The steps involved in our approach are presented in the following subsections[21].

G. Web Crawling

The analysis of the structure and informatics of the web is facilitated by a data collection technique known as Web Crawling. The collection of as many beneficiary web pages as possible along their interconnection links in a speedy yet proficient manner is the prime intent of crawling[22]. Automatic traversal of web sites, downloading documents and tracing links to other pages are some of the features of a web crawler program. Numerous search engines utilize web crawlers for gathering web pages of interest besides indexing them.

H. Stop Words Removal.

It is necessary and beneficial to remove the commonly utilized stop words such as "it", "can" ,"an" y "and", "by", "for", "from", "of', "the", "to", "with" and more either while parsing a document to obtain information about the content or while scoring fresh URLs that the page recommends[19]. This procedure is termed as stop listing. Stop listing aids in the reduction of size of the indexing file besides enhancing efficiency and value.

I. Stemming Algorithm

Variant word forms in Information Retrieval are restricted to a common root by Stemming. The postulation lying behind is that, two words possess the same root represent identical concepts. Thus terms possessing to identical meaning yet appear morphologically dissimilar are identified in an IR system by matching query and document terms with the aid of Stemming[21]. Stemming facilitates the reduction of all words possessing an identical root to a single one. This is achieved by removing each word of its derivational and inflectional suffixes.

J. Keywords Representation

We possess the distinct keywords and their counts in each of the each crawled web page as a result of stemming. These keywords are then represented in a form to ease the process of near duplicates detection. This representation will reduce the search space for the near duplicate detection. Initially the keywords and their number of occurrences in a

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

web page have been sorted in descending order based on their counts[23]. Afterwards, n numbers of keywords with highest counts are stored in a table and the remaining keywords are indexed and stored in another table[20] [24]. In our approach the value of n is set to be 4. The similarity score between two documents can be calculated if and only the prime keywords of the two documents are similar. Thus the search space is reduced for near duplicates detection.

K. Similarity Score Calculation

If the prime keywords of the new web page do not match with the prime keywords of the pages in the table, then the new web page is added in to the repository[25]. If all the keywords of both pages are same then the new page is considered as duplicate and thus is not included in the repository. If the prime keywords of new page are same with a page in the repository, then the similarity score between the two documents is calculated.

IV. RESULTS AND DISCUSSIONS

Through the web is a huge information store, various features such as the presence of huge volume of unstructured or semi-structured data; their dynamic nature; existence of duplicate and near duplicate documents and the like pose serious difficulties for Information Retrieval. The voluminous amount of web documents swarming the web have posed a huge challenge to the web search engines making them render results of less relevance to the users.

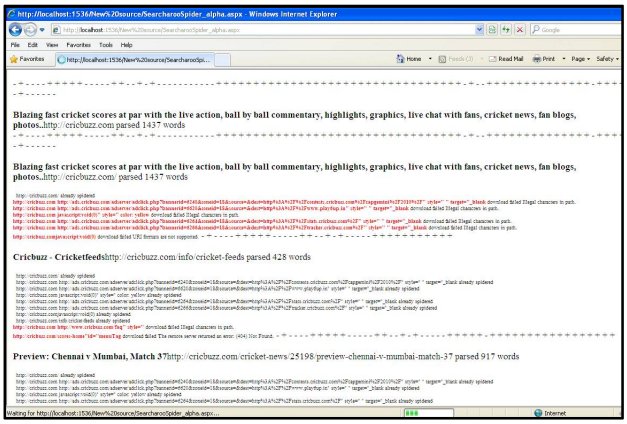

Fig.1.Extracted result

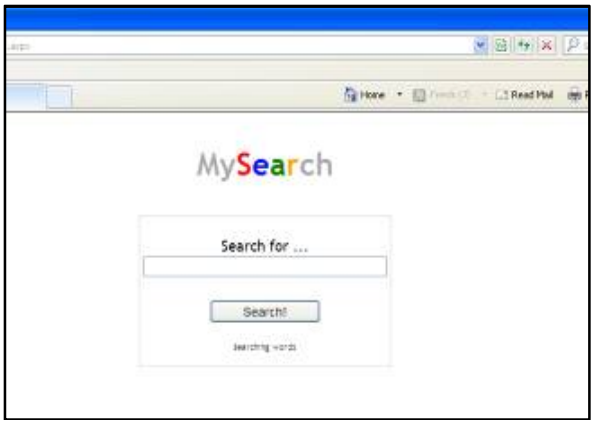

Fig. 2. Search a keyword result from extracted details

The detection of duplicate and near duplicate web documents in web crawling. The proposed approach has detected the duplicate and near duplicate web pages efficiently based on the keywords extracted from the web pages. Furthermore, reduced memory spaces for web repositories and improved search engine quality have been accomplished through the proposed duplicates detection approach. The extracted result is shown in Fig.1. The searching of a keyword result from the extracted details is depicted in Fig.2.The search result is shown in Fig.3. The end of the search result is depicted in Fig.4.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

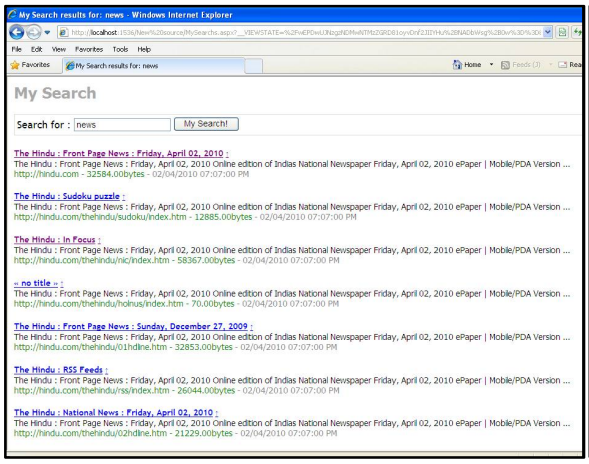

Fig. 3.Search Result

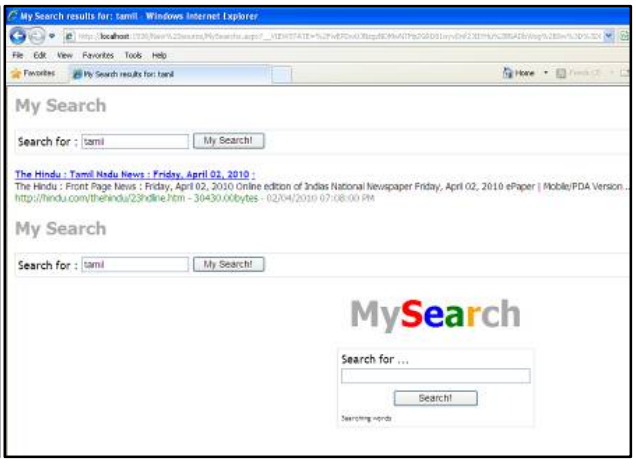

Fig 4. End of search result

V. CONCLUSION AND FUTURE WORK

The experiments highlight that one of the most important factors for efficient and accurate indexing for record linkage and de-duplication is the proper definition of blocking keys. Because training data in the form of true matches and true no matches is often not available in real-world applications, it is commonly up to domain and linkage experts to decide how such blocking keys are defined. The number of true matched candidate record pairs generated by the different techniques, but also large differences for several indexing techniques depending upon the setting of their parameters. The implementation of further recently developed new indexing techniques into the Fibril framework, as well as the investigation of learning techniques for efficient and accurate indexing. The indexing techniques presented in this survey are heuristic approaches that aim to split the records in a database (or databases) into (possibly overlapping) blocks such that matches are inserted into the same block and non matches into different blocks. While future work in the area of indexing for record linkage and de-duplication should include the development of more efficient and more scalable new indexing techniques, the ultimate goal of such research will be to develop techniques that generate blocks such that it can be proven that 1) all comparisons between records within a block will have a certain minimum similarity with each other, and 2) the similarity between records in different blocks is below this minimum similarity.

REFERENCES

1. C.W. Kelman, J. Bass, and D. Holman, "Research Use of Linked Health Data—A Best Practice Protocol", Australian NZ J. Public Health, vol. 26, pp. 251-255, 2002.
2. W.E. Winkler, "Overview of Record Linkage and Current Research Directions", Technical Report RR2006/02, US Bureau of the Census, 2006.
3. J. Jonas and J. Harper, "Effective Counterterrorism and the Limited Role of Predictive Data Mining", Policy Analysis, no. 584, pp. 1-11, 2006.
4. H. Hajishirzi, W. Yih, and A. Kolcz, "Adaptive Near-Duplicate Detection via Similarity Learning", Proc. 33rd Int'l ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR '10), pp. 419-426, 2010.
5. W. Su, J. Wang, and F.H. Lochovsky, "Record Matching over Query Results from Multiple Web Databases", IEEE Trans. Knowledge and Data Eng., vol. 22, no. 4, pp. 578-589, Apr. 2010.
6. M. Bilenko, S. Basu, and M. Sahami, "Adaptive Product Normalization: Using Online Learning for Record Linkage in Comparison Shopping", Proc. IEEE Int'l Conf. Data Mining (ICDM '05), pp. 58-65, 2005.
7. P. Christen and K. Goiser, "Quality and Complexity Measures for Data Linkage and Deduplication", Quality Measures in Data Mining, ser. Studies in Computat.ional Intelligence, F. Guillet and H. Hamilton, eds., vol. 43, Springer, pp. 127-151, 2007.
8. A. Aizawa and K. Oyama, "A Fast Linkage Detection Scheme for Multi-Source Information Integration", Proc. Int'l Workshop Challenges in Web Information Retrieval and Integration (WIRI '05), 2005.
9. I. Bhattacharya and L. Getoor, "Collective Entity Resolution in Relational Data", ACM Trans. Knowledge Discovery from Data, vol. 1, no. 1, pp. 5-es, 2007.
10. P. Christen, R. Gayler, and D. Hawking, "Similarity-Aware Indexing for Real-Time Entity Resolution", Proc. 18th ACM Conf. Information and Knowledge Management (CIKM '09), pp. 1565-1568, 2009.

ISSN(Online): 2320-9801
ISSN (Print) : 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 3, March 2016

11. S.E. Whang, D. Menestrina, G. Koutrika, M. Theobald, and H. Garcia-Molina, "Entity Resolution with Iterative Blocking", Proc. 35th ACM SIGMOD Int'l Conf. Management of Data (SIGMOD '09), pp. 219-232, 2009.
12. P. Christen, "Febrl: An Open Source Data Cleaning, Deduplication and Record Linkage System With a Graphical User Interface", Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '08), pp. 1065-1068, 2008.
13. L. Gu and R. Baxter, "Decision Models for Record Linkage", Selected Papers from AusDM, LNCS 3755, Springer, 2006.
14. P. Christen, "Automatic Record Linkage Using Seeded Nearest Neighbour and Support Vector Machine Classification", Proc. 14th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '08), pp. 151-159, 2008.
15. C. Xiao, W. Wang, and X. Lin, "Ed-Join: An Efficient Algorithm for Similarity Joins with Edit Distance Constraints", Proc. VLDB Endowment, vol. 1, no. 1, pp. 933-944, 2008.
16. Y. Zhang, X. Lin, W. Zhang, J. Wang, and Q. Lin, "Effectively Indexing the Uncertain Space", IEEE Trans. Knowledge and Data Eng., vol. 22, no. 9, pp. 1247-1261, Sept. 2010.
17. T. Bernecker, H.-P.Kriegel, N. Mamoulis, M. Renz, and A. Zuefle, "Scalable Probabilistic Similarity Ranking in Uncertain Databases", IEEE Trans. Knowledge and Data Eng., vol. 22, no. 9, pp. 1234-1246, Sept. 2010
18. D. Dey, V. Mookerjee, and D. Liu, "Efficient Techniques for Online Record Linkage", IEEE Trans. Knowledge and Data Eng., vol. 23, no. 3, pp. 373-387, Mar. 2011.
19. G.V. Moustakides and V.S. Verykios, "Optimal Stopping: A Record-Linkage Approach", J. Data and Information Quality, vol. 1, pp. 9:1-9:34, 2009.
20. A. Behm, S. Ji, C. Li, and J. Lu, "Space-Constrained Gram-Based Indexing for Efficient Approximate String Search", Proc. IEEE Int'l Conf. Data Eng. (ICDE '09), pp. 604-615, 2009.
21. P. Christen and A. Pudjijono, "Accurate Synthetic Generation of Realistic Personal Information", Proc. 13th Pacific-Asia Conf. Advances in Knowledge Discovery and Data Mining (PAKDD '09), vol. 5476, pp. 507-514, 2009.
22. T. de Vries, H. Ke, S. Chawla, and P. Christen, "Robust Record Linkage Blocking Using Suffix Arrays and Bloom Filters", ACM Trans. Knowledge Discovery from Data, vol. 5, no. 2, pp. 1-27, 2011.
23. M. Weis, F. Naumann, U. Jehle, J. Lufter, and H. Schuster, "Industry-Scale Duplicate Detection", Proc. VLDB Endowment, vol. 1, no. 2, pp. 1253-1264, 2008.
24. M. Bilenko, B. Kamath, and R.J. Mooney, "Adaptive Blocking: Learning to Scale up Record Linkage", Proc. Sixth Int'l Conf. Data Mining (ICDM '06), pp. 87-96, 2006.
25. M. Michelson and C.A. Knoblock, "Learning Blocking Schemes for Record Linkage", Proc. 21st Nat'l Conf. Artificial Intelligence (AAAI '06), 2012.

BIOGRAPHY

M.Prem Kumarreceived his MCA from Bharathiar University in the year 2008. He completed his M.Phil (part time) in Computer Science at SreeSaraswathiThyagaraja College, Pollachi in the year 2012. He is currently working as The Head of Department, Department of Computer Technology at SreeSaraswathiThyagaraja College, Pollachi, Coimbatore District, TN, India. He is doing a Minor Research Project funded by UGC, Hyderabad. His areas of interests are Data Mining, Object Oriented Programming and Web Development.

H. VigneshRamamoorthy received his M.Sc (Software Engineering) from Dr.Mahalingam College of Engineering and Technology (affiliated to Anna University, Chennai) in 2010, MCA from Bharathiar University in 2012, Post Graduate Diploma in Mobile Computing from Annamalai University in 2012, Post Graduate Diploma in English Language Teaching from Annamalai University in 2013. He completed his M.Phil (part time) in Computer Science at SreeSaraswathiThyagaraja College, Pollachi in the year 2013. He is currently working as an Assistant Professor of Computer Science atSreeSaraswathiThyagaraja College, Pollachi, Coimbatore District, TN, India. He is also pursuing his part time Ph.D in Bharathiar University. His areas of interests are Data Mining, Wireless Sensor Networks and Mobile Computing.