RESEARCH ARTICLE OPEN ACCESS

Automatic Detection of Name and Aliases From The Web

R.V.Gandhi [1], N.Suman [2], Md.Zuber [3], U.Mahender [4]

Abstract

Many celebrities and experts from various fields may have been referred by not only their personal names but also by their aliases on web. Aliases are very important in information retrieval to retrieve complete information about a personal name from the web, as some of the web pages of the person may also be referred by his aliases. The aliases for a personal name are extracted by previously proposed alias extraction method. In information retrieval, the web search engine automatically expands the search query on a person name by tagging his aliases for complete information retrieval thereby improving recall in relation detection task and achieving a significant mean reciprocal rank (MRR) of search engine. For the further substantial improvement on recall and MRR from the previously proposed methods, our proposed method will order the aliases based on their associations with the name using the definition of anchor texts-based co-occurrences between name and aliases in order to help the search engine tag the aliases according to the order of associations. The association orders will automatically be discovered by creating an anchor texts-based co-occurrence graph between name and aliases. Ranking support vector machine (SVM) will be used to create connections between name and aliases in the graph by performing ranking on anchor texts-based co-occurrence measures. The hop distances between nodes in the graph will lead to have the associations between name and aliases. The hop distances will be found by mining the graph. The proposed method will outperform previously proposed methods, achieving substantial growth on recall and MRR.

Keywords: Automated Discovery, Aliases, lexical patterns, Co-occurrences, Pattern Extraction, Alias Extraction

I. INTRODUCTION

The aliases for a personal name are extracted by previously proposed alias extraction method. In information retrieval, the web search engine automatically expands the search query on a person name by tagging his aliases for complete information retrieval thereby improving recall in relation detection task and achieving a significant mean reciprocal rank (MRR) of search engine. For the further substantial improvement on recall and MRR from the previously proposed methods, our proposed method will order the aliases based on their associations with the name using the definition of anchor texts-based co-occurrences between name and aliases in order to help the search engine tag the aliases according to the order of associations. The association orders will automatically be discovered by creating an anchor texts-based co-occurrence graph between name and aliases. Ranking support vector machine (SVM) will be used to create connections between name and aliases in the graph by performing ranking on anchor texts-based co-occurrence measures. Retrieving information about people from web search engines can become difficult when a person has nicknames or name aliases. For example, a newspaper article on the baseball player might use the real name, Hideki Matsui, whereas a blogger would use the alias, Godzilla, in a blog entry.

Identification of entities on the web is difficult for two fundamental reasons:

1. Different entities can share the same name (i.e., lexical ambiguity).
2. A single entity can be designated by multiple names (i.e., referential ambiguity).

For example, the lexical ambiguity consider the name Jim Clark. Aside from the two most popular namesakes, the formula-one racing champion and the founder of Netscape. Referential ambiguity occurs because people use different names to refer to the same entity on the web.

The problem of referential ambiguity of entities on the web has received much less attention. In this paper, the authors examine on the problem of automatically extracting the various references on the web of a particular entity. The contributions can be summarized as follows:

1. Propose a lexical pattern-based approach to extract aliases of a given name using snippets returned by a web search engine.

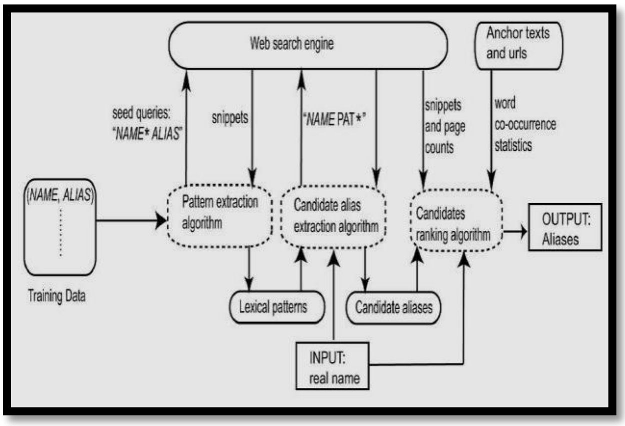
The lexical patterns are generated automatically using a set of real world name alias data. Evaluate the confidence of extracted lexical patterns and retain the patterns that can accurately discover aliases for various personal names.

2. To select the best aliases among the extracted candidates, the authors propose numerous ranking scores based upon three approaches: lexical pattern frequency, word co-occurrences in an anchor text graph, and page counts on the web.

3. Train a ranking support vector machine to learn the optimal combination of individual ranking scores to construct a robust alias extraction method. Conduct a series of experiments to evaluate the various components of the proposed method on three data sets: An English personal names data set, An English place names data set and A Japanese personal names data set.

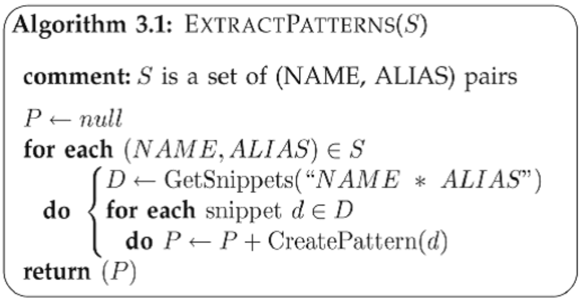
II. RELATED WORK

Alias identification: It is closely related to the problem of cross-document coreference resolution in which the objective is to determine whether two mentions of a name in different documents refer to the same entity. Baggaand Baldwin [10] proposed a cross-document coreference resolution algorithm by first performing within document coreference resolution for each individual document to extract coreferencechains. In personal name disambiguation the goal is to disambiguate various people that share the same name (namesakes) [3], [4]. However, the name disambiguation problem differs fundamentally from that of alias extraction. Because in name disambiguation the objective is to identify the different entities that are referred by the same ambiguous name; in alias extraction, the authors are interested in extracting all references to a single entity from the web. Approximate string matching algorithms have been used for extracting variants or abbreviations of personal names [11]. Bilenkoand Mooney [12] proposed a method to learn a string similarity measure to detect duplicates in bibliography databases. However, an inherent limitation of such string matching approaches is that they cannot identify aliases.



Extracting Lexical Patterns from Snippets:

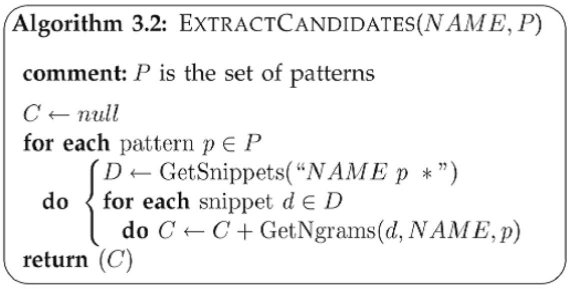
Extracting Lexical Patterns from Snippets for names and aliases, snippets convey useful semantic clues that can be used to extract lexical patterns that are frequently used to express aliases of a name. The authors use the wildcard operator * to perform a NEAR query and it matches with one or more words in a snippet. Propose the shallow pattern extraction method illustrated in Fig. 3 to capture the various ways in which information about aliases of names is expressed on the web. From each snippet, the **Create-Pattern** function extracts the sequence of words that appear between the name and the alias.



```
Algorithm 3.1: EXTRACTPATTERNS($S$)
  comment: $S$ is a set of (NAME, ALIAS) pairs
  $P \leftarrow null$
  for each $(NAME, ALIAS) \in S$
    do $D \leftarrow$ GetSnippets("$NAME * ALIAS$")
       for each snippet $d \in D$
         do $P \leftarrow P +$ CreatePattern($d$)
  return ($P$)
```

Fig. 3. Given a set of (NAME, ALIAS) instances, extract lexical patterns.

Given a name, NAME and a set, Pof lexical patterns, the function **ExtractCandidates** returns a list of candidate aliases for the name.



```
Algorithm 3.2: EXTRACTCANDIDATES($NAME, P$)
  comment: $P$ is the set of patterns
  $C \leftarrow null$
  for each pattern $p \in P$
    do $D \leftarrow$ GetSnippets("$NAME\ p\ *$")
       for each snippet $d \in D$
         do $C \leftarrow C +$ GetNgrams($d, NAME, p$)
  return ($C$)
```

Fig. 4. Given a name and a set of lexical patterns, extract candidate aliases.

Lexical Pattern Frequency:

If the personal name under consideration and a candidate alias occur in many lexical patterns, then it can be considered as a good alias for the personal name. Consequently, rank a set of candidate aliases in the descending order of the number of different lexical patterns in which they appear with a name.

Hub Discounting:

A frequently observed phenomenon related to the web is that many pages with diverse topics link to so-called hubs such as Google, Yahoo, or MSN. Two anchor texts might link to a hub for entirely different reasons. Therefore, co-occurrences coming from hubs are prone to noise. If the majority of anchor texts linked to a particular web site use the real name to do so, then the confidence of that page as a source of information regarding the person whom we are interested in extracting aliases increases.

Training:

Using a data set of name-alias pairs, train a ranking support vector machine to rank candidate aliases according to their strength of association with a name. For a name-alias pair, we define three types of features: anchor text-based co-occurrence measures, web page-count-based association measures and frequencies of observed lexical patterns. The nine co-occurrence measures are computed with and without weighting for hubs to produce 18(2 x 9) features. The four page-count-based association measures. The frequencies of lexical patterns extracted by Algorithm 3.1 are used as features in training the ranking SVM. Normalize each measure to range [0,1] to produce feature vectors for training. The trained SVM model can then be used to assign a ranking score to each candidate alias. Finally, the highest-ranking candidate is selected as the correct alias of the name.

Data Set:

Create three name-alias data sets: The English personal names data set (50 names), The English place names data set (50 names) and The Japanese personal names (100 names) data set. Aliases were manually collected after referring various information sources such as Wikipedia and official home pages. A website might use links for purely navigational purposes, which do not convey any semantic clues. In order to remove navigational links in the data set, prepare a list of words that are commonly used in navigational menus such as *top*, last, next, previous, links, etc., and ignore anchor texts that contain these words. Remove any links that point to pages within the same site. Data set contains 24,456,871 anchor texts pointing to 8,023,364 urls.

Pattern Extraction:

Algorithm 3.1 extracts over 8,000 patterns for the 50 English personal names data set. Rank the patterns according to their F-scores to identify the patterns that accurately convey information about aliases.

$$\text{Precision}(S) = \frac{\text{No. of correct aliases retrieved by s}}{\text{No. of correct aliases retrieved by s}}$$

$$\text{Recall}(S) = \frac{\text{No. of correct aliases retrieved by s}}{\text{No. of correct aliases retrieved by s}}$$

$$\text{F}(S) = \frac{2 * \text{Precision}(s) * \text{Recall}(s)}{\text{bPrecision}(s) + \text{Recall}(s)}$$

Alias Extraction:

Mean reciprocal rank (MRR) and AP [26] is used to evaluate the different approaches. MRR is defined as follows:

$$\text{MRR} = \frac{1}{\text{n}} \sum\nolimits_{\text{i}=1}^{n} \frac{1}{Ri}$$

Among the numerous individual ranking scores, the best results are reported by the hub weighted fidfscore (tfidf(h)). It is noteworthy for anchor text-based ranking scores, the hub-weighted version always outperforms the non hub-weighted counterpart, which justifies the proposed hub-weighting method. With each data set, we performed a five-fold cross validation. The proposed method reports high scores for both MRR and AP on all three data sets.

III. SYSTEM ANALYSIS

The first step in finding an appropriate solution to a given problem is to understand the problem and its domain. The main objective of the analysis is to capture a complete, unambiguous, and consistent picture of the requirements of the system and what the system must do to satisfy the user's requirements and needs. This is accomplished by constructing several models of the system that concentrate on describing what the system does rather than how it does it. Separating the behavior of a system from the way that behavior is implemented requires viewing the system from the perspective of the user rather than that of the machine. Analysis is the process of transforming a problem definition from a fuzzy set of facts and myths into a coherent statement of a system's requirements.

EXISTING SYSTEM:

The existing namesake disambiguation algorithm assumes the real name of a person to be given and does not attempt to disambiguate people who are referred only by aliases.

LIMITATIONS OF EXISTING SYSTEM

- To low MRR and AP scores on all data sets.
- To complex hub discounting measure.

PROPOSED SYSTEM:

The proposed method will work on the aliases and get the association orders between name and aliases to help search engine tag those aliases according to the orders such as first order associations, second order associations etc so as to substantially increase the recall and MRR of the search engine while searching made on person names. The term recall is defined as the percentage of relevant documents that were in fact retrieved for a search query on search engine. The mean reciprocal rank of the search engine for a given sample of queries is that the average of the reciprocal ranks for each query. The term word co-occurrence refers to the temporal property of the two words occurring at the same web page or same document on the web. The anchor text is the clickable text on web pages, which points to a particular web document. Moreover the anchor texts are used by search engine algorithms to provide relevant documents for search results because they point to the web pages that are relevant to the user queries. So the anchor texts will be helpful to find the strength of association between two words on the web. The anchor texts-based co-occurrence means that the two anchor texts from the different web pages point to the same the URL on the web. The anchor texts which point to the same URL are called as inbound anchor texts. The proposed method will find the anchor texts-based co-occurrences between name and aliases using co-occurrence statistics and will rank the name and aliases by support vector machine according to the co-occurrence measures in order to get connections among name and aliases for drawing the word co-occurrence graph. Then a word co-occurrence graph will be created and mined by graph mining algorithm so as to get the hop distance between name and aliases that will lead to the association orders of aliases with the name. The search engine can now expand the search query on a name by tagging the aliases according to their association orders to retrieve all relevant pages which in turn will increase the recall and achieve a substantial MRR.

2.4.1 Keyword Extraction Algorithm

Matsuo, Ishizuka proposed a method called keyword extraction algorithm that applies to a single document without using a corpus. Frequent terms are extracted first, and then a set of co-occurrences between each term and the frequent terms, i.e., occurrences in the same sentences, are generated. Co-occurrence distribution showed the importance of a term in the document**.** However, this method only extracts a keyword from a document but not correlate any more documents using anchor texts-based co-occurrence frequency**.**

MODULES:

1. Co-occurrences in Anchor Texts
2. Role of Anchor Texts
3. Anchor Texts Co-occurrence Frequency
4. Ranking Anchor Texts
5. Discovery of Association Orders

MODULE DESCRIPTION:

1. Co-occurrences in Anchor Texts

The proposed method will first retrieve all corresponding URLs from search engine for all anchor texts in which name and aliases appear. Most of the search engines provide search operators to search in anchor texts on the web. For example, Google provides In anchor or Allinanchor search operator to retrieve URLs that are pointed by the anchor text given as a query. For example, query on “Allinanchor:Hideki Matsui” to the Google will provide all URLs pointed by Hideki Matsui anchor text on the web.

2. Role of Anchor Texts

The main objective of search engine is to provide the most relevant documents for a user’s query. Anchor texts play a vital role in search engine algorithm because it is clickable text which points to a particular relevant page on the web. Hence search engine considers anchor text as a main factor to retrieve relevant documents to the user’s query. Anchor texts are used in synonym extraction, ranking and classification of web pages and query translation in cross language information retrieval system.

3. Anchor Texts Co-occurrence Frequency

The two anchor texts appearing in different web pages are called as inbound anchor texts if they point to the same URL. Anchor texts co-occurrence frequency between anchor texts refers to the number of different URLs on which they co-occur. For example, if p and x that are two anchor texts are co-occurring, then p and x point to the same URL. If the co-occurrence frequency between p and x is that say an example k, and then p and x co-occur in k number of different URLs. For example, the picture of Arnold Schwarzenegger is shown in Fig 2 which is being liked by four different anchor texts. According to the definition of co-occurrences on anchor texts, Terminator and Predator are co-occurring. As well, The Expendables and Governator are also co-occurring.

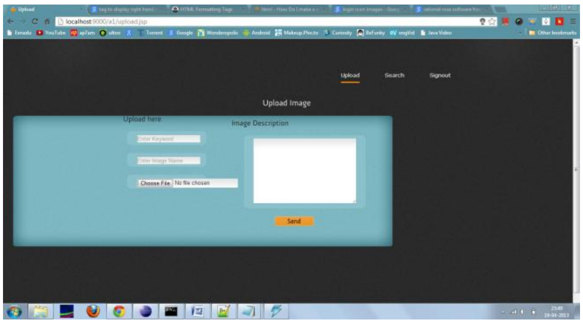
4. Ranking Anchor Texts

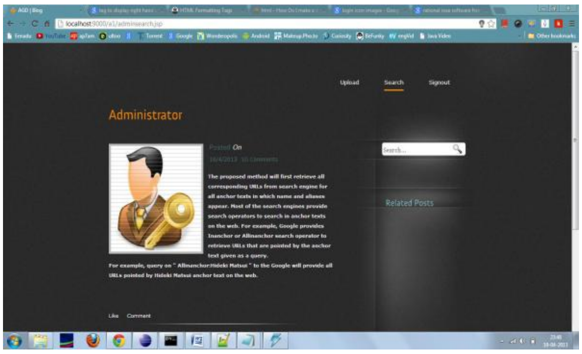
Ranking SVM will be used for ranking the aliases. The ranking SVM will be trained by training samples of name and aliases. All the co-occurrence measures for the anchor texts of the training samples will be found and will be normalized into the range of [0-1]. The normalized values termed as feature vectors will be used to train the SVM to get the ranking function to test the given anchor texts of name and aliases. Then for each anchor text, the trained SVM using the ranking function will rank the other anchor texts with respect to their co-occurrence measures with it. The highest ranking anchor text will be elected to make a first–order association with its corresponding anchor text for which ranking was performed. Next the word co-occurrence graph will be drawn for name and aliases according to the first order associations between them.

5. Discovery of Association Orders

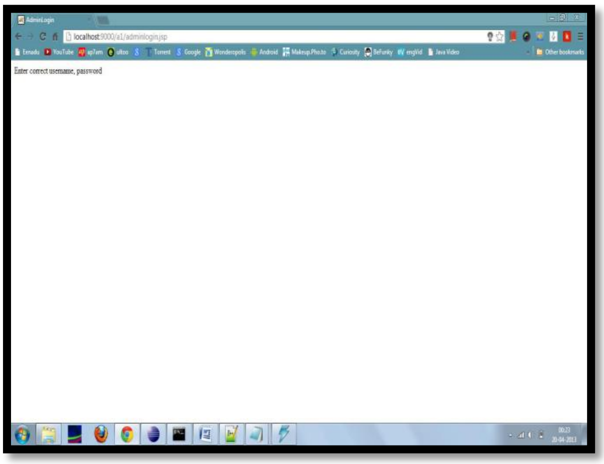
Using the graph mining algorithm, the word co-occurrence graph will be mined to find the hop distances between nodes in graph. The hop distances between two nodes will be measured by counting the number of edges in-between the corresponding two nodes. The number of edges will yield the association orders between two nodes. According to the definition, a node that lies n hops away from p has an n-order co-occurrence with p. Hence the first, second and higher order associations between name and aliases will be identified by finding the hop distances between them. The search engine can now expand the query on person names by tagging the aliases according to the association orders with the name. Thereby the recall will be substantially improved by 40% in relation detection task. Moreover the search engine will get a substantial MRR for a sample of queries by giving relevant search results.

IV. RESULTS


localhost:9000/a1/upload.jsp


localhost:9000/a1/adminsearch.jsp

The proposed method will first retrieve all corresponding URLs from search engine for all anchor texts in which name and aliases appear. Most of the search engines provide search operators to search in anchor texts on the web. For example, Google provides Inanchor or Allinanchor search operator to retrieve URLs that are pointed by the anchor text given as a query.
For example, query on " Allinanchor:Hideki Matsui " to the Google will provide all URLs pointed by Hideki Matsui anchor text on the web.


localhost:9000/a1/adminlogin.jsp

Enter correct username, password

V. CONCLUSION

The proposed method will compute anchor texts-based co-occurrences among the given personal name and aliases, and will create a word co-occurrence graph by making connections between nodes representing name and aliases in the graph based on their first order associations with each other. The graph mining algorithm to find out the hop distances between nodes will be used to identify the association orders between name and aliases. Ranking SVM will be used to rank the anchor texts according to the co-occurrence statistics in order to identify the anchor texts in the first order associations. The web search engine can expand the query on a personal name by tagging aliases in the order of their associations with name to retrieve all relevant results thereby improving recall and achieving a substantial MRR compared to that of previously proposed methods. Proposed a lexical-pattern-based approach to extract aliases of a given name. The candidates are ranked using various ranking scores computed using three approaches: lexical pattern frequency, co-occurrences in anchor texts, page counts-based association measures. Construct a single ranking function using ranking support vector machines. The proposed method reported high MRR and AP scores on all three data sets and outperformed numerous baselines and a previously proposed alias extraction algorithm. Discounting co-occurrences from hubs is important to filter the noise in co-occurrences in anchor texts. The extracted aliases significantly improved recall in a relation detection task and render useful in a web search task.

REFERENCES

[1] G. Salton and M. McGill, Introduction to Modern Information Retreival. McGraw-Hill Inc., 1986.

[2] M. Mitra, A. Singhal, and C. Buckley, “Improving Automatic Query Expansion,” Proc. SIGIR ’98, pp. 206-214, 1998.

[3] P. Cimano, S. Handschuh, and S. Staab, “Towards the Self-Annotating Web,” Proc. Int’l World Wide Web Conf. (WWW’04), 2004.

[4] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, Takeda, K. Hasida, and M. Ishizuka, “Polyphonet: An Advanced Social Network Extraction System,” Proc. WWW ’06, 2006.

[5] P. Turney, “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews,” Proc. Assoc. for Computational Linguistics (ACL’02), pp. 417-424, 2002.

[6] A. Bagga and B. Baldwin, “Entity-Based Cross-Document Coreferencing Using the Vector Space Model,” Proc. Int’l Conf. Computational Linguistics (COLING ’98), pp. 79-85, 1998.

[7] C. Galvez and F. Moya-Anegon, “Approximate Personal Name- Matching through Finite-State Graphs,” J. Am. Soc. For Information Science and Technology, vol. 58, pp. 1-17, 2007.

[8] M. Bilenko and R. Mooney, “Adaptive Duplicate Detection Using Learnable String Similarity Measures,” Proc. SIGKDD ’03, 2003.

[9] T. Hokama and H. Kitagawa, “Extracting Mnemonic Names of People from the Web,” Proc. Ninth Int’l Conf. Asian Digital Libraries.

[10] G. Mann and D. Yarowsky, “Unsupervised Personal Name Disambiguation,” Proc. Conf. Computational Natural Language Learning (CoNLL’03), 2003. (C : 206).

[11] R. Bekkermanand A. McCallum, “Disambiguating Web Appearances of People in a Social Network,” Proc. Int’l World Wide Web Conf. (WWW ’05), pp. 463-470, 2005. (C : 166).

[12] A. Baggaand B. Baldwin, “Entity-Based Cross-Document Coreferencing using the Vector Space Model,” Proc. Int’l Conf. Computational Linguistics (COLING ’98), pp. 79-85, 1998. (C : 240).

[13] C. Galvez and F. Moya-Anegon, “Approximate Personal Name Matching through Finite-State Graphs,” J. Am. Soc. for Information Science and Technology, vol. 58, pp. 1-17, 2007. (C : 26)

[14] Automatic Discovery of Personal Name Aliases from the Web Presenter : Chen, Zhong-Yong by Danushka Bollegala, The University of Tokyo, Tokyo Yutaka Matsuo, The University of Tokyo, Tokyo Mitsuru Ishizuka, The University of Tokyo, Tokyo.

[15] T. Kudo, K. Yamamoto, and Y. Matsumoto, “Applying Conditional Random Fields to Japanese Morphological Analysis,” Proc. Conf. Empirical Methods in Natural Language (EMNLP ’04), 2004. (C : 106).

Author’s Profile

Mr. R.V.GANDHI, Post Graduated in Computer Science & Engineering (M.Tech) , Jawaharlal Nehru Technological University Hyderabad , 2009 and Bachelor of Technology (B.Tech) in Computer Science & Engineering, Jawaharlal Nehru Technological University, Hyderabad , 2007. He is working presently as an Assistant Professor in Department of Computer Science & Engineering in **Mother Theresa Institute of Engineering and Technology**, Melumoi, Palamaner, Chitoor Dist, A.P, INDIA. He has 4+ years Experience.

Mr. Nalugotla Suman working as an Assistant Professor in the Department of Computer Science and Engineering at **Hi-Point college of Engineering and Technology**, Hyderabad, RR dist Andhra Pradesh, India. He has received M.Tech in Software Engineering from JNTUH, Hyderabad, Andhra Pradesh, India. He has 6 years of teaching experience. His research interests are Data Mining, Software Engineering, Computer Networks, High Performance Computing and Cloud Computing.

Mr. Md.Zuber received Doctorate in Philosophy (Ph.d) in Computer science and Engineering from Allahabad Central University, Allahabad. U.P,. He has received his post graduation (M.Tech) from JNTUH, Hyderabad, and Andhra Pradesh. He is working as Assistant Professor in the Department of Computer Science and Engineering at Addis Ababa University in Ethiopia. He has totally 6years of Teaching Experience. His research interests are Image Processing, Data mining Networks, Pervasive Applications and High Performance Computing.

Mr. U.MAHENDER, Post Graduated in Computer Science & Engineering (M.Tech) ,HOLY MARY INSTITUTE OF TECHNOLOGY AND SCIENCE, Jawaharlal Nehru Technological University Hyderabad , In 2010 . And Bachelor Of Technology (B.TECH), in Vidya Bharathi Institute Of Technology, Jawaharlal Nehru Technological University, Hyderabad, in 2007. He is working presently as an Assistant Professor in Department of Computer Science & Engineering in **TKR COLLEGE OF ENGINEERING AND TECHNOLOGY** medbowli, meerpet, RR Dist, and A.P, INDIA. He has 3+ years Experience.