Maximizing the Search Engine Efficiency

V. KISHORE, V. SARITHA, G. RAMA MOHAN RAO, M. RAMAKRISHNA

Abstract—Due to the massive amount of heterogeneous information on the web, insufficient and vague user queries, and use of the different query by different users for same aims, the information retrieval process deals with a huge amount of uncertainty and doubt. Under such circumstances, designing an efficient retrieval function and ranking algorithm by which the most relevant results are provided is of the greatest importance.

Now a day’s search engines became more vital in retrieving information about unknown entities. In the world an entity may be recognized by more than on name (alias). For example many celebrities and experts from various fields may have been referred by not only their original 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.

In recent years, researchers have been developing algorithms for the automatic information retrieval to meet the demands of retrieving almost all data; 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 contribution presented in this paper makes human intervention one step more down by outperform previously proposed methods, achieving substantial improvement on recall and MRR.


I. INTRODUCTION

There is a huge quantity of text, audio, video, and other documents available on the Internet, on about any subject. Users need to be able to find relevant information to satisfy their particular information needs. There are two ways of searching for information: to use a search engines or to browse directories organized by categories. There is still a large part of the Internet (private databases and intranets) that is not accessible.

The organization of this paper is as follows. We briefly mention the search engines history, features, and services. We present the generic architecture of a search engine. We discuss its Web crawling component, which has the task to collect WebPages to be indexed. Then we focus on the Information Retrieval component which has the task of retrieving documents (mainly text documents) that answer a user query. We describe information retrieval techniques, focusing on the challenges faced by search engines. One particular challenge is the large scale, given by the huge number of WebPages available on the Internet. Another challenge is inherent to any information retrieval system that deals with text: the ambiguity of the natural language (English or other languages) that makes it difficult to have perfect matches between documents and user queries. We propose methods used to evaluate the performance of the Information Retrieval component by overcoming the ambiguity of name aliases. 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.

II. OVER VIEW OF SEARCH ENGINES

2.1 Search engines:

There are many general-purpose search engines available on the Web. A resource containing up to date information on the most used search engines is: www.searchenginewatch.com Here are some popular search engines AltaVista Google Yahoo! MSN Search. Meta-search engines combine several existing search engines in order to provide documents relevant to a user query. Their task is reduced to ranking results from the different search engines and eliminating duplicates. Some examples are: www.metacrawler.com, www.mamma.com, and www.dogpile.com.

2.2 Search engine History

The very first tool used for searching on the Internet was called Archie (the name stands for "archive"). It was created in 1990 by Alan Emtage, a student at McGill University in Montreal. The program downloaded the directory listings of all the files located on public anonymous FTP sites, creating a searchable database of filenames. Gopher was created in 1991 by Mark McCahill at the University of Minnesota. While Archie indexed file names, Gopher indexed plain text documents.

Soon after, many search engines appeared and became popular. These included Excite, Infoseek, Inktomi, Northern Light, and AltaVista. In some ways, they competed with...
popular directories such as Yahoo. Later, the directories integrated or added on search engine technology for greater functionality.

Around 2001, the Google search engine rose to prominence (Page and Brin, 1998). Its success was based in part on the concept of link popularity and PageRank that uses the premise that good or desirable pages are pointed to by more pages than others. Google's minimalist user interface was very popular with users, and has since spawned a number of imitators. In 2005, it indexed approximately 8 billion pages, more than any other search engine. It also offers a growing range of Web services, such as Google Maps and online automatic translation tools...etc

In 2002, Yahoo acquired Inktomi and in 2003, Overture, which owned AlltheWeb and AltaVista. Despite owning its own search engine, Yahoo initially kept using Google to provide its users with search results. In 2004, Yahoo launched its own search engine based on the combined technologies of its acquisitions and providing a service that gave preeminence to the Web search engine over its manually-maintained subject directory.

MSN Search is a search engine owned by Microsoft, which previously relied on others for its search engine listings. In early 2005 it started showing its own results, collected by its own crawler. Many other search engines tend to be portals that merely show the results from another company's search engine.

2.3 Search Engine Architectures

The components of a search engine are: Web crawling (gathering WebPages), indexing (representing and storing the information), retrieval (being able to retrieve documents relevant to user queries), and ranking the results in their order of relevance. Figure 1 presents a simplified view of the components of a search engine.

Figure 1: The simplified architecture of a search engine.

2.4 Web crawler

Web crawling is an integral piece of infrastructure for search engines. The web crawler is a program that automatically traverses the web by downloading the pages and following the links from page to page [Koster 1999]. A general purpose of web crawler is to download any web page that can be accessed through the links.

Generic crawlers [1, 2] crawl documents and links belonging to a variety of topics, whereas focused crawlers [3, 4, 5] use some specialized knowledge to limit the crawl to pages pertaining to specific topics. For web crawling, issues like freshness and efficient resource usage have previously been addressed [6, 7, 8]. However, the problem of elimination of near-duplicate web documents and returning all the pages related to user query (including aliases) in a generic crawl has not received attention.

Documents that are exact duplicates of each other (due to mirroring and plagiarism) are easy to identify by standard checksum techniques. A more difficult problem is the identification of near-duplicate documents. Two such documents are identical in terms of content but differ in a small portion of the document such as advertisements, counters and timestamps. These differences are irrelevant for web search. So if a newly-crawled page $P_{\text{duplicate}}$ is deemed a near-duplicate of an already-crawled page $P$, the crawl engine should ignore $P_{\text{duplicate}}$ and its entire out-going links (intuition suggests that these are probably near-duplicates of pages reachable from $P$). Elimination of near-duplicates saves network bandwidth, reduces storage costs and improves the quality of search indexes. It also reduces the load on the remote host that is serving such web pages.

A system for detection of near-duplicate pages faces a number of challenges. First and foremost is the issue of scale: search engines index billions of web-pages; this amounts to a multi-terabyte database. Second, the crawl engine should be able to crawl billions of web-pages per day. So the decision to mark a newly-crawled page as a near-duplicate of an existing page should be made quickly. Finally, the system should use as few machines as possible.

III. OVERVIEW OF INFORMATION RETRIEVAL

3.1 Information Retrieval:

The last few decades have witnessed the birth and explosive growth of the web. It is indisputable that the exponential growth of the web has made it into a huge ocean. Finding web pages with high quality is one of the important purposes in search engines. The quality of pages is defined based on request and preferences of user. Usually, there are millions of relative pages with each query.

Information storage and retrieval systems make large volumes of text accessible to people with information needs (van Rijsbergen, 1979; Salton and McGill, 1983). A person (hereafter referred to as the user) approaches such a system with some idea of what they want to find out, and the goal of the system is to fulfill that need. The person provides an outline of their requirements—perhaps a list of keywords relating to the topic in question, or even an example document. The system searches its database for documents that are related to the user’s query, and presents those that are most relevant. Nevertheless, users specially consider only 10 to 20 of those pages. The basic architecture is as shown in Figure 2.

Figure 2: Information Retrieval System - Basic Architecture
Queries transform the user’s information need into a form that correctly represents the user’s underlying information requirement and is suitable for the matching process. Query formatting depends on the underlying model of retrieval used (Boolean models [Bookstein, 1985], Vector Space models [Salton & McGill, 1983], Probabilistic models [Maron & Kuhns, 1960; Robertson, 1977], based on artificial intelligence techniques [Maaeng, 1992; Evans 1993]).

A matching algorithm matches a user’s requests (in terms of queries) with the document representations and retrieves documents that are most likely to be relevant to the user. A matching algorithm addresses two issues:

1. How to decide how well documents match a user’s information request. Blair & Maron [1985] showed that it is very difficult for users to predict the exact words or phrases used by authors in desired documents. Hence if a document term does not match search terms then a relevant document may not be retrieved.

2. Another issue involved in matching is how to decide the order in which the documents are to be shown to the user. Typically the matching algorithms calculate a matching number for each document and retrieve the documents in the decreasing order of this number.

The user rates documents presented as either relevant or non-relevant to his/her information need. The basic problem facing any IR system is how to retrieve only the relevant documents for the user’s information requirements, while not retrieving non-relevant ones. Various system performance criteria like precision and recall have been used to gauge the effectiveness of the system in meeting users’ information requirements.

Precision is defined as the ratio of the number of relevant retrieved documents to the total number of retrieved documents.

\[
P = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of retrieved documents}}
\]

Recall is the ratio of the number of relevant retrieved documents to the total number of relevant documents available in the document collection.

\[
R = \frac{\text{Number of relevant documents retrieved}}{\text{Total number of relevant documents}}
\]

Using this information, a retrieval system is able to rank the documents that match and present the most relevant ones to the user first—a major advantage over the standard Boolean approach which does not order retrieved documents in any significant way. From a list of documents sorted by predicted relevance, a user can select several from the top as being most pertinent. In a system using the Boolean model, users would have to search through all the documents retrieved, determining each one’s importance manually.

Systems that employ simple query methods usually provide other facilities to aid users in their search. For example, a thesaurus can match words in a query with those in the index that have similar meaning. The most effective tool exploited by information retrieval systems is user feedback. Instead of having users re-engineer their queries in an adhoc fashion; a system that measures their interest in retrieved documents can adjust the queries for them. By weighting terms in the query according to their performance in selecting appropriate documents, the system can refine a query to reflect the user’s needs more precisely. The user is required only to indicate the relevance of each document presented. Finally, an information retrieval system’s conceptual model defines how documents in its database are compared.

### 3.2 Evaluation of Information Retrieval Systems

To compare the performance of information retrieval systems there is a need for standard test collections and benchmarks. The TREC forum (Text Retrieval Conference) provides test collections and organizes competition between IR systems every year, since 1992. In order to compute evaluation scores, we need to know the expected solution. Relevance judgments are produced by human judges and included in the standard test collections.

CLEF (Cross-Language Evaluation Forum) is another evaluation forum that organizes competition between IR systems that allow queries or documents in multiple languages, since the year 2000.

In order to evaluate the performance of an IR system we need to measure how far down the ranked list of results will a user need to look to find some or all the relevant documents.

### 3.3 IR Paradigms

This section briefly describes various research paradigms prevalent in IR and where our work fits in. At a broad level, research in IR can be categorized [Chen, 1995] into three categories: Probabilistic IR, Knowledge based IR, and IR based on machine learning techniques.

#### 3.3.1. Probabilistic IR:

Probabilistic retrieval is based on estimating a probability of relevance of a document to the user for the given user query. Typically relevance feedback from a few documents is used to establish the probability of relevance for other documents in the collection [Fuhr et al., 1991; Gordon, 1988]. There are three different learning strategies used in probabilistic retrieval. Estimation of probabilities of relevance is done for a set of sample documents [Robertson & Sparck Jones, 1976], or a set of sample queries [Maron & Kuhns, 1960] and extended to all the documents or queries. Inference networks [Turtle & Croft, 1990] use a document and query network that capture probabilistic dependencies among the nodes in the network.

#### 3.3.2. Knowledge based IR:

This approach focuses on modeling two areas. First, it tries to model the knowledge of an expert retriever in terms of the expert’s domain knowledge, i.e. user search strategies and feedback heuristics. An example of such an approach is the Unified Medical Language System. Another area that has been modeled is the user of the system. This typically follows the way the librarian develops a client profile. Although knowledge based approaches might be effective in certain domains, it may not be applicable in all domains [Chen et al., 1991].

#### 3.3.3. Learning systems based IR:

This approach is based on algorithmic extraction of knowledge or identifying patterns in the data. There are three broad areas within this approach: Symbolic Learning, Neural Networks and Evolution based algorithms. In the Symbolic Learning approach knowledge discovery is done typically by inductive learning by creating a hierarchical arrangement of concepts and producing IF-THEN type production rules. ID3 decision-making algorithm [Quinlan, 1986] is one such popular algorithm.
IV. RELATED WORK

4.1 Keyword Extraction Algorithm:

Keyword extraction algorithm that applies to a single document without using a corpus was proposed by Matsuo, Ishizuka [10]. 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 do not correlate any more documents using anchor text-based co-occurrence frequency.

4.2 Transitive Translation Approach:

Transitive translation approach to find translation equivalents of query terms and constructing multilingual lexicons through the mining of web anchor texts and link structures was proposed by Lu, Chien and Lee [11]. The translation equivalents of a query term can be extracted via its translation in an intermediate language. However this method did not associate anchor texts using the definition of co-occurrences.

4.3 Feature Selection Method:

A novel feature selection method based on part-of-speech and word co-occurrence was proposed by Liu, Yu, Deng, Wang, Bian[12]. According to the components of Chinese document text, they utilized the word’s part-of-speech attributes to filter lots of meaningless terms. Then they defined and used co-occurrence words by their part-of-speech to select features. The results showed that their method can select better features and get a more pleasant clustering performance. However, this method does not use anchor texts-based co-occurrences on words.

4.4 Data Treatment Strategy:

A data treatment strategy to generate new discriminative features, called compound-features(c-features) for the sake of text classification was proposed by Figueiredo et al. [13]. These c-features are composed by terms that co-occur in documents without any restrictions on order or distance between terms within a document. This strategy precedes the classification task, in order to enhance documents with discriminative c-features. This method extracts only a keyword from a document but not correlate any more documents using anchor texts.

<table>
<thead>
<tr>
<th>Anchor Texts</th>
<th>x</th>
<th>C - {x}</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>k</td>
<td>n-k</td>
<td>N</td>
</tr>
<tr>
<td>V-{P}</td>
<td>K-k</td>
<td>N-n-k+K</td>
<td>N-n</td>
</tr>
<tr>
<td>V-{P}</td>
<td>k</td>
<td>N-K</td>
<td>N</td>
</tr>
</tbody>
</table>

Table 1. Contingency Table for anchor Texts 'p' and 'x'.

4.5 Alias Extraction Method:

A method to extract aliases from the web for a given personal name was first proposed by Bollegala, Matsuo, and Ishizuka [9]. They have used lexical pattern approach to extract candidate aliases. The incorrect aliases have been removed by page counts, anchor text co-occurrence frequency, and lexical pattern frequency. However, this method considered only the first order co-occurrences on aliases to rank them but did not focus on the second order co-occurrences to improve recall and achieve a substantial MRR for the web search engine.

V. PROPOSED METHOD

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 Mean Reciprocal Rank (MRR) of the search engine while searching made on person names.

The MRR applies to the binary relevance task: for a given query, any returned document is labeled “relevant” or “not relevant”, and if r_i is the rank of the highest ranking relevant document for the i^th query, then the reciprocal rank measure for that query is 1/r_i and the MRR is just the 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 URL on the web. The anchor texts which point to the same URL are called as inbound anchor texts [9]. 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 (SVM) according to the co-occurrence measures in order to get connections among name and aliases for drawing the word co-occurrence graph.

![Proposed method - overview](image.png)

Figure 3: Proposed method - overview

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.

The proposed method is outlined in Figure 3 and comprises four main components namely

- Computation of word co-occurrence statistics
- Ranking anchor texts
- Creation of anchor text co-occurrence graph
- Discovery of association orders

To compute anchor texts-based co-occurrence measures, there are nine co-occurrence statistics [9] used in anchor text mining to measure the associations between anchor texts:

- Co-occurrence Frequency (CF)
- Term Frequency–Inverse Document Frequency (tfidf)
- Chi Square (CS)
- Log Likelihood Ratio (LLR)
- Point wise Mutual Information (PMI)
- Hyper Geometric Distribution (HG)
- Cosine
- Overlap
- Dice

Ranking Support Vector Machine will be used to rank the anchor texts with respect to each anchor text to identify the highest ranking anchor text for making first order associations among anchor texts.

### 5.1 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 Inanchor or Allinanchor search operator to retrieve URLs that are pointed by the anchor text given as a query. For example, query on “Allinanchor: Sachin Tendulkar” to the Google will provide all URLs pointed by Sachin Tendulkar anchor text on the web.

**Figure 4: A picture of Sachin Tendulkar being linked by different anchor texts on the web**

Next the contingency table will be created as described in Table 1 for each pair of anchor texts to measure their strength. There in

- p and x: two input anchor texts.
- C: set of input anchor texts except p
- V: set of all words that appear in anchor texts
- C-{p}: anchor texts except p
- V-{p}: anchor texts except p
- K: co-occurrence frequency between p and x
- n: sum of the co-occurrence frequencies between p and all anchor texts in C.
- N: sum of the co-occurrence frequencies between all words in V and all anchor texts in C.

#### 5.1.1 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.

### 5.2 Anchor Texts Co-occurrence Frequency

The two anchor texts appearing in different web pages are called as inbound anchor texts [9] if they point to the same URL. Anchor texts co-occurrence frequency [9] 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 Sachin Tendulkar is shown in Figure 4 which is being liked by six different anchor texts. According to the definition of co-occurrences on anchor texts, Little Master and Master Blaster are co-occurring. As well, The Sachin and Tendulkar are also co-occurring.

#### 5.2.1 Co-occurrence Frequency (CF)

The CF [9] is the simplest measurement among all and it denotes the value of k in the Table 1.

#### 5.2.2 Term Frequency–Inverse Document Frequency (tfidf)

The CF is biased towards highly frequent words. But tfidf [9, 14] resolves the bias by reducing the weight, that is, assigned to the words on anchor texts. The tfidf score for the anchor texts p and x is calculated from Table 1 as

\[
\text{tfidf}(p, x) = k \log \left( \frac{N}{K+k} \right) \quad \text{...(1)}
\]

#### 5.2.3 Chi Square (CS)

The Chi Square [9] is used to test the dependence between two words in natural language processing tasks. Given the contingency table in Table 1, the \(X^2\) measure compares the observed frequency in Table 1 with the expected frequency for independence. Then it is likely that the anchor texts p and x are dependent if the difference between the observed and expected frequencies is large. The \(X^2\) measure sums the difference between the observed and expected frequencies and is scaled by the expected values. The \(X^2\) measure is given as

\[
X^2 = \sum_{ij} \frac{(O_{ij} - E_{ij})^2}{E_{ij}} \quad \text{...(2)}
\]

Where \(O_{ij}\) and \(E_{ij}\) are the observed and expected frequencies respectively. Using Equation (2), the \(X^2\) score for anchor texts p and x from the Table 1 is as follows

\[
\text{CS}(p, x) = N\left[\frac{(N-K-n+k)(n-k)(K-k)}{n(K-N-K)(N-n)}\right]^{2} \quad \text{...(3)}
\]

#### 5.2.4 Log Likelihood Ratio (LLR)

LLR [9, 15] is the ratio between the likelihoods of two alternative hypotheses: that the texts p and x are independent

---

**Figure 3: Overview of the proposed method**

Sachin Tendulkar

- Little master
- Master Blaster
- God of Cricket
- Tendulkar
- Tendlya

---

**Table 1.**

<table>
<thead>
<tr>
<th>p</th>
<th>x</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>V</td>
</tr>
<tr>
<td>C-{p}</td>
<td>V-{p}</td>
</tr>
<tr>
<td>K</td>
<td>n</td>
</tr>
<tr>
<td>N</td>
<td>N</td>
</tr>
</tbody>
</table>

---

All Rights Reserved © 2012 IJARCSEE
or they are dependent. LLR is calculated using the Table 1 as follows

\[
\text{LLR}(p, x) = k \log \frac{n_k}{n} + \left(\log \frac{(n-k)N}{m(N-K)} + (K-k) \log \frac{N(K-k)}{K(n-K)} \right) + \left(\log \frac{N(K-n-k)}{(N-K)(N-n)} \right) \quad (4)
\]

5.1.3.5 Point wise Mutual Information (PMI)

PMI [9, 16] reflects the dependence between two probabilistic events. The PMI is defined for y and z events as

\[
\text{PMI}(y, z) = \log_2 \left( \frac{P(y,z)}{P(y)P(z)} \right) \quad (5)
\]

Where \(P(y)\) and \(P(z)\), respectively, represent the probability of events y and z. Whereas \(P(y, z)\) is the joint probability of y and z. The PMI is calculated from Table 1 as

\[
\text{PMI}(y, z) = \log_2 \left( \frac{k}{n} \right) \quad (6)
\]

5.1.3.6 Hyper Geometric Distribution (HG)

Hyper Geometric distribution [9, 16] is a discrete probability distribution that represents the number of successes in a sequence of draws from a finite population without replacement. For example, the probability of the event that “k red balls are contained among n balls, which are arbitrarily selected from among \(N\) balls containing \(K\) red balls” is given by the hyper geometric distribution hg(N, K, n, k) as

\[
hg(N, K, n, k) = \frac{\binom{K}{k} \binom{N-K}{n-k}}{\binom{N}{n}} \quad (7)
\]

The hyper geometric distribution is applied to the values of Table 1 and the HG (p, x) is computed as the probability of observing more than \(k\) number of co-occurrences of p and x.

\[
\text{HG}(p, x) = \log_2 \left( \sum_{i=k}^{N-k} \text{hg}(N, K, n, i) \right)
\]

\[
\max \left\{ 0, N+k-n \right\} \geq l \geq \min \left\{ n, K \right\} \quad (8)
\]

5.1.3.7 Cosine

Cosine [9] computes the association between anchor texts. The association between elements in two sets X and Y is computed as

\[
\text{cosine}(x, y) = \frac{|X \cap Y|}{\sqrt{|X| \sqrt{|Y|}}} \quad (9)
\]

Where \(|X|\) represents the number of elements in set X. Considering X be the co-occurrences of anchor texts x and Y be the co-occurrences of anchor text p, then cosine measure from Table 1 is computed as

\[
\text{cosine}(p, x) = \frac{k}{\sqrt{n} \sqrt{K}} \quad (10)
\]

5.1.3.8 Overlap

The overlap [9] between two sets X and Y is defined as

\[
\text{overlap}(x, y) = \frac{|X \cap Y|}{\min \{|X|,|Y|\}} \quad (11)
\]

Assuming that X and Y, respectively, represent occurrences of anchor texts p and x. The overlap of (p, x) to evaluate the appropriateness is defined as

\[
\text{Overlap}(p, x) = \frac{k}{\min\{n,K\}} \quad (12)
\]

5.1.3.9 Dice

Dice [9, 17] retrieves collocations from large textual corpora. The Dice is defined over two sets X and Y as

\[
\text{Dice}(X, Y) = \frac{2|X \cap Y|}{|X| + |Y|} \quad (13)
\]

\[
\text{Dice}(p, x) = \frac{2k}{n + K} \quad (14)
\]

5.2 Ranking Anchor Texts

Ranking SVM [9, 18] 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 association between them.

Figure 5: Word Co-occurrence graph for a personal name “Sachin Tendulkar”

5.3 Word Co-occurrence Graph

Word co-occurrence graph is an undirected graph where the nodes represent words that appear in anchor texts on the web. For each word in anchor text, a node will be created in the graph. According to the definition of co-occurrences if the two anchor texts co-occur in pointing to the same URL, then undirected edge will be drawn between them to denote their co-occurrences. A word co-occurrence graph like that shown in Fig 3 will be created for the name and aliases according to their first order associations among them. Each name and aliases will be represented by a node in the graph. The two nodes will be connected if they make first order associations between them. The edge between nodes will describe that the nodes bearing anchor texts co-occur according to the definition of anchor texts co-occurrences. Next the hop distance between nodes will be identified in order to have first, second, and higher order associations between name and aliases by graph mining algorithm.

5.4 Discovery of Association Orders

Using the graph mining algorithm [19, 20], the word co-occurrence graph will be mined to find the hop distances

Sachin Tendulkar

Little Master

Member of Parliament

Mumbai Indians

Cricket

Sports

Master

Blaster

Master

Sports

Figure 5: Word Co-occurrence graph for a personal name “Sachin Tendulkar”

5.3 Word Co-occurrence Graph

Word co-occurrence graph is an undirected graph where the nodes represent words that appear in anchor texts on the web. For each word in anchor text, a node will be created in the graph. According to the definition of co-occurrences if the two anchor texts co-occur in pointing to the same URL, then undirected edge will be drawn between them to denote their co-occurrences. A word co-occurrence graph like that shown in Fig 3 will be created for the name and aliases according to their first order associations among them. Each name and aliases will be represented by a node in the graph. The two nodes will be connected if they make first order associations between them. The edge between nodes will describe that the nodes bearing anchor texts co-occur according to the definition of anchor texts co-occurrences. Next the hop distance between nodes will be identified in order to have first, second, and higher order associations between name and aliases by graph mining algorithm.

5.4 Discovery of Association Orders

Using the graph mining algorithm [19, 20], the word co-occurrence graph will be mined to find the hop distances

Sachin Tendulkar

Little Master

Member of Parliament

Mumbai Indians

Cricket

Sports

Master

Blaster

Master

Sports

Figure 5: Word Co-occurrence graph for a personal name “Sachin Tendulkar”

5.3 Word Co-occurrence Graph

Word co-occurrence graph is an undirected graph where the nodes represent words that appear in anchor texts on the web. For each word in anchor text, a node will be created in the graph. According to the definition of co-occurrences if the two anchor texts co-occur in pointing to the same URL, then undirected edge will be drawn between them to denote their co-occurrences. A word co-occurrence graph like that shown in Fig 3 will be created for the name and aliases according to their first order associations among them. Each name and aliases will be represented by a node in the graph. The two nodes will be connected if they make first order associations between them. The edge between nodes will describe that the nodes bearing anchor texts co-occur according to the definition of anchor texts co-occurrences. Next the hop distance between nodes will be identified in order to have first, second, and higher order associations between name and aliases by graph mining algorithm.

5.4 Discovery of Association Orders

Using the graph mining algorithm [19, 20], 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 lays 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 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.

5.5 Data Set

To train and evaluate the proposed method, there are two data sets: the personal names data set and the place names data set. The personal names data set includes people from various fields of cinema, sports, politics, science, and mass media. The place names data set contains aliases for US states.

VI. 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.

REFERENCES


V.KISHORE M.Tech, (Ph.D), working as Assoc. Professor & HOD of Department of Computer Science & Engineering, SANA Engineering College, Kodad, Nalgonda. He has Published 4 International Journals, His areas of interest is Network Security, Information Retrieval Systems…etc.,

V.SARITHA, she is pursuing M.Tech (CSE) in SANA Engineering College, Kodad, Nalgonda, Andhra Pradesh, India.

G. RAMAMOHAN RAO, M.Tech. working as Asst. Professor in Department of Computer Science and Engineering, SANA Engineering College, Kodad, Nalgonda.

M. RAMAKRISHNA, he is pursuing M.Tech (CSE) in SANA Engineering College, Kodad, Nalgonda, Andhra Pradesh, India.