Final version of the accepted paper. Cite as: *T. Anwar and M. Abulaish, Namesake Alias Mining on the Web and its Role Towards Suspect Tracking, Information Sciences, 276(0), pp. 123-145, 2014.*

Namesake Alias Mining on the Web and its Role Towards Suspect Tracking

Tarique Anwar[a,*], Muhammad Abulaish[a,b,**]

[a] *Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia*
[b] *Department of Computer Science, Jamia Millia Islamia (A Central University), New Delhi, India*

Abstract

With the proliferation of social media, the number of active web-users is rapidly increasing these days. They create and maintain their personal web-profiles, and use them to interact with others in the cyber-space. Currently two major problems are being faced to automatically identify these web-users and correlate their web-profiles. First is the presence of *namesakes* on the Web, and the second is the use of *alias names*. In this paper, we propose a context-based text mining approach to discover alias names for all the namesakes sharing a common name on the Web, and leave the task of selecting the namesake of interest on part of the user. The proposed method employs a search-engine API to retrieve relevant webpages for a given name. The retrieved webpages are modeled into a graph, and a clustering algorithm is applied to disambiguate the webpages. Thereafter each obtained cluster standing for a namesake is mined for alias identification following a text pattern based statistical technique. The existing research works do not consider the presence of namesakes on the Web to mine aliases, which is impractical. The novelty of the proposed approach lies in discovering this drawback of existing works. Additionally the contribution includes the disambiguation technique that doesn't need to have a pre-determined number of clusters to be generated and the light-weight text pattern based alias mining technique. The number of clusters in the proposed method is rather determined dynamically by the inflation parameter, the pre-determination of which is comparatively much easier. Experimental results on different components demonstrate the robustness of the proposed alias mining approach. This paper also brings forth the significance of alias mining to the problem of suspect monitoring and tracking on the Web.

Keywords: Web content mining, Web people search, Alias mining, Namesake disambiguation, Clustering.

1. Introduction

With the widespread digitization of printed materials and the cheap and easy accessibility of Internet, the Web is growing rapidly both in scope as well as in depth [25]. Since last few years, the newer generation people are undergoing through a great revolution in their lives adopting several recent trends [33, 27]. They find the Web as helpful, interesting and entertaining to interact with others whom they know well in their real life, and very often they even don't know. Sometimes these interactions are intended to perform some personal tasks (e.g. *online-shopping* to buy something needed). Quite often they are just to have some entertainment by exchanging thoughts with different people. Social media (e.g. *Facebook*, *Twitter*, *Youtube*, various weblogs, and so on) have become an important part of their lives. Creating and maintaining personal webpages are among the other activities which keep them intact with WWW. According to the statistics of *DoubleClick Ad Planner*[1], social media sites are amongst the top ranking websites with largest number of visitors as well as largest number of page views. During the month of April in 2011, *Facebook* has remained on top with 880,000,000 unique visitors and 910,000,000,000 page views. Table 1 presents a list of top 10 websites visited in this month, along with the number of unique visitors and number of pages navigated. These numbers show the importance of their move towards social media. Thus, in addition to the real world, they have started being in a virtual world of WWW, which can be said as a reflection of their real world. The user-generated contents resulting from their frequent interactions with the Web has made it ever since the largest repository of electronically accessible data with a potential to reveal a lot of undiscovered crucial information about users' online behaviors and activities [42, 47, 28, 3]. These online activities could further be used to infer their real life activities and involvements. However, due to the unstructured and unorganized nature of the available data, it is a challenging task to retrieve and integrate all these information collected from diversified source. Search engines, like *Google* and *Yahoo!*, make use of keywords of the given search query to find matches on the Web, and return a ranked set of webpages. For people search, this simple method is ineffective. Machine learning and data mining techniques need to be applied further on the returned results of search engines to analyze the information, and deduce some meaningful and relevant information about web-user profiles and activities.

As said in previous works [26, 11], Web people search has

[*] Currently Tarique Anwar is a PhD Candidate in the Web and Data Engineering research group at Swinburne University of Technology, and a member of the Victoria Research Lab (VRL), National Information and Communication Technologies Australia (NICTA), Melbourne, Australia.

[**] To whom correspondence should be made. Telefax: +91-11-26980014.

Email addresses: `tAnwar@swin.edu.au` (Tarique Anwar), `abulaish@ieee.org` (Muhammad Abulaish)

[1] `http://www.google.com/adplanner/`

Table 1: Most popular websites in April, 2011

Website	Category	Visitiors	Page Views
facebook.com	Social Network	880 000 000	910 000 000 000
youtube.com	Online Video	800 000 000	100 000 000 000
yahoo.com	Web Portal	660 000 000	77 000 000 000
live.com	Search Engine	550 000 000	36 000 000 000
wikipedia.org	Encyclopedia	490 000 000	7 000 000 000
msn.com	Web Portal	450 000 000	15 000 000 000
blogspot.com	Blogging service	410 000 000	5 400 000 000
baidu.com	Search Engine	340 000 000	110 000 000 000
bing.com	Search Engine	340 000 000	11 000 000 000
microsoft.com	Software	340 000 000	2 700 000 000

become very common. Around 30 percent of Web search queries account for person names. However, due to the unstructured nature of Web contents, many ambiguities exist in the returned results. For the task of Web people search, there are two major problems currently being faced. *i*) The first one is that, as the Web is a common unit of global access, just like the real world, multiple persons sharing the same name called *namesakes*, exist on the Web as well. It becomes difficult to consider them as different individuals. For example, on passing a search query on Google for the text "*tarique anwar*" to find exact matches, the top 20 returned results consist of pages referring to 12 different individuals with this name and it has no such functionality to mark them as of different persons. The Web is also having a dominating nature for famous personalities. As there is no such person with the name *tarique anwar* so popular, the results are highly varied. At the same time, a search for "*azim premji*", the famous Indian business tycoon and the chairman of Wipro Technologies, returned webpages with 18 out of top 20 referring to this individual directly. The remaining two referred to *azim premji university* and *azim premji foundation* which indirectly refer to the same person. *ii*) The second problem is that, again like the real world, quite often a single person is known by multiple names on the Web called *alias names* or *mnemonic names*, and it becomes difficult to relate all the pages referring to the same individual by different alias names. For example, *Albert Einstein* on the Web is also known as the *father of modern science*, *Albert* and *Alby*. Sometimes these alternate names are used just because of their simplicity (e.g. *Alby*), or sometimes to highlight any specific aspect (*father of modern science*). Quite often they are also used by the person to represent himself or herself to a specific group of people who know the original identity, while it remains hidden to the rest of the public (e.g. using a nickname while discussing with others through any social media platform or chat server). In addition to these intentionally used alias names, it's quite common for the person to misspell a name which produces a different lexical structure than the actual name. Whatever be the reason of use, these aliases are of great importance to gather facts for a specific person. They can increase its scope of search by expanding the query after personalizing it [12], which in turn will increase its recall. Some other applications of aliases are using them in the form of metadata for Web entities to annotate them [13, 48], for disambiguating Web entities [14] by annotating them by aliases, identifying relationships among entities in social media [32], and analyzing sentiments from comments on social media [18, 34] by including alias names to identify the person.

In this paper, we propose a context-based approach to mine alias names of persons from the vast electronically accessible data on the WWW, taking into account the issue of namesakes. The system starts working with retrieving target webpages using Google API. A graph-based clustering technique is then applied on the retrieved pages to group them into different clusters, where each cluster is expected to correspond to a specific namesake. Thereafter, webpages from each of the clusters are processed by a light-weight alias mining algorithm to extract aliases for each namesake. The novelty of the proposed approach lies in application of the clustering technique for namesake disambiguation, the alias mining algorithm, and their integration to sort out person name ambiguities on the Web.

2. Related Work

In this section, we present a brief review of web-mining and its applications to identify aliases on the Web. According to Kosala and Blockeel [25], web mining tasks can be categorized into three major areas- *web content mining*, *web usage mining*, and *web structure mining*. In addition to its huge size, the Web is also characterized by its dynamic and diverse nature, which calls for a continuous treatment in the due course of time. Research on this area has gained adequate attention of researchers bringing forth solutions to various kinds of novel problems, which range from recommender systems [15, 52] and personalized search [43] to problems like spam filtering [51, 41] and the question-answering systems [29, 31]. The ambiguous nature of the Web is found a major hindrance in them. Very often different entities are designated by the same name and also vice versa (i.e., a single entity is designated by multiple names). In the first case, there exist entities on webpages that could stand for multiple meanings. For example, the word *Puma*, it can be a Brazilian brand of *sports car*, or *Puma AG* as a German shoe and sportswear company or *Cagaur*, a large cat, *AMD Puma* as a mobile computing platform, *Lake Puma Yumko* as a lake in Tibet, Puma as a *local language* of Nepal, person whose *surname* is Puma and many more. This ambiguity needs to be resolved for refining the entity. The area of *named entity disambiguation* deals with this problem, and a considerable number of efforts have been made in the past [14, 17]. These techniques make use of the context of its surroundings to resolve the conflict and select the best suited interpretation among the several possible. In contrast to this, the latter case needs to find out the alternate names that are being used on the Web to represent the real name. Three problems analogous to alias identification are named entity recognition [38] that deals with identification of entity names (e.g. name of a person), cross-document coreference resolution [8] that finds out if entity names from different documents are inter-related in any way, and word sense disambiguation [40] that identifies the exact sense or meaning of a word in reference to the surrounding context. Person names are one of the several kinds of entities and their alias mining uses the concept inherited from these problems.

Lexical variants of person names are very common on the Web. For example, the name *Albert Einstien* may appear as *A. Einstein* or *Einstein Albert*. In [36], Piskorski *et al.* used some string distance metrics to acquire suffix-based lemmatization patterns for person name matching in Polish language that can also be generalized.

In [20], Hokama and Kitagawa proposed a pattern-based system for alias mining from Web documents in Japanese language. They assumed that wherever an alias name of a person appears on the Web with his or her real name, it would be in the lexical pattern "aliasName *koto*[2] realName". For a given real name they performed a search for the query "* *koto* realName" to extract candidate aliases, which are then ranked by looking into the text patterns surrounding the real name and the candidate.

Although the approach of Hokama and Kitgawa [20] was a good initiative, the presence of few drawbacks made Bollegala *et al.* to criticize them in [11]. The word *koto* is actually an ambiguous word in Japanese that could mean many other words too whose English translations are *incident*, *thing*, *matter*, *experience* and *task*. It may often result in producing either incorrect or noisy aliases. Bollegala *et al.* mined a set of lexical text patterns for English textual data using a test dataset of real name and alias name pairs, that are generally used to map the real name of a person to his or her alias name. In addition, the accuracy of each pattern is also computed using the metric *F-score*. They used these text patterns along with real names to generate unified search queries, which are passed on the Web to extract candidate aliases. To rank the candidates, a support vector machine (SVM) is trained on a set of 23 features. 18 of them are computed from an anchor text graph mined from webpage anchor texts, 4 are the different association measures between the real name and a candidate, and the last one is frequency of the lexical patterns used to find the candidate.

In our earlier work [6], based on the work of Bollegala *et al.* [11] we proposed a light-weight pattern based approach intended to serve for crime suspect investigations on the Web. Using the top ten text patterns mined by Bollegala *et al.*, we extracted the candidates and then calculated three feature values to capture three distinct information for each of them. Finally, candidates are ranked on the unified value.

We found that all the existing methods for alias mining to the best of our knowledge follow a common approach to find target webpages, that is by passing Web search queries comprising real name and a set of text patterns. They consider the target person as the only individual with the input name to exist on the Web, which practically can rarely happen for most of the cases. They mix up information from all the retrieved pages regardless of the presence of namesakes. The practical applicability of the methodology seems to be very weak unless the queried person is highly dominant on the Web over all other namesakes. In contrast, our context-based approach takes care of each of the multiple namesakes sharing the given real name and mines alias names separately for each of them.

3. Proposed Approach

As discussed in section 1, person name ambiguities are common on the Web and becoming more prominent each day with the growing number of Web users. The standard Web search engines do not provide much support to overcome the challenges posed by them for people search. At the current stage, manual efforts are employed to disambiguate the retrieved results and identify the exact person, one is looking for. It is quite a tedious job to go through the vast set of results returned by search engines. In a broad perspective, the proposed approach is intended to address both the problems associated with Web people search mentioned earlier, which are resolving person name ambiguities and integration of all alias names referring to a specific person. Figure 1 presents functional overview of the proposed approach, showing its four major tasks as *data crawling and pre-processing*, *entity extraction*, *namesake identification*, and *alias identification*. The first module described in subsection 3.1 gathers the data to be mined, whereas the second and third modules described in subsections 3.3 and 3.4 respectively work for disambiguating namesake profiles, and finally the last module described in subsection 3.5 mines aliases for identified namesakes.

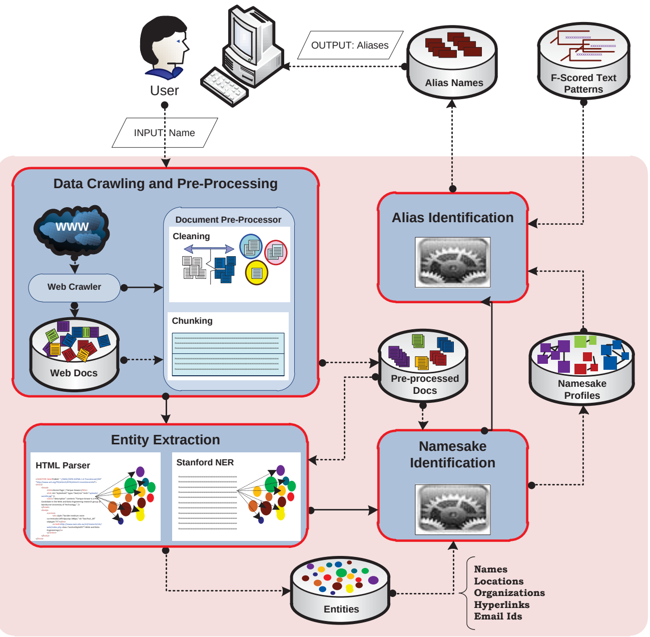


Figure 1: Architecture of the proposed system

3.1. Data Crawling and Pre-processing

To start with, the Web is queried using Google API to search for pages containing the input name. Generally, we find it very common for a person to use different forms of his or her name (e.g., full name, abbreviated name, primary name). So, in order to retrieve all pages containing the different possible forms of a given name, the query passed to the Web is not limited to exact matches, rather results are retrieved even for the partial

[2] It's a text in Japanese language, when translated into English, it means *"be called"*.

matches[3]. On the other hand, as the Web is unstructured up to a large extent, relaxing the criteria for partial match leads to the retrieval of a large number of irrelevant pages. Very often the contents of these pages are found to be full of advertisements or contain many incomplete sentences, which generally increase the false positive and false negative rates of the named entity recognizer (NER). To overcome this problem, we apply some heuristic rules to restrict incomplete sentences for further considerations, like exclusion of text blocks containing less than a predefined number of words, say less than 5 words, which is a sign of an incomplete sentence and lack of required information. Hence the prime objective of this task is concerned with two major issues – first one is to consider all the relevant pages, and the other one is to discard all those representing noise. To transform the unstructured textual data into machine usable form, some sort of cleaning and chunking is done as a pre-processing task. These pre-processed documents are used by the following modules of *entity extraction* and *namesake identification* to generate namesake web-profiles by disambiguating the gathered data.

3.2. Namesake Profile Disambiguation

After the set of cleaned webpages are collected for a person name, say p, a graph-based clustering technique is applied to group them into different clusters, where each cluster of webpages is expected to stand for a namesake sharing the common name. In the present scenario, because of no support from search engines to differentiate between the namesake individuals, manual efforts are employed by the Web users. In our research study, we tried to gather all these factors that lead a human being to differentiate between namesakes manually and finally arrived at three major factors summarized below:

- *Content Overlapping*: Generally, different webpages referring to the same person contain a set of common facts in them, in the form of entities that may be the name of other persons to whom the target person is anyhow related (e.g. network partners, family members, and fellow worker), the organization names with which he or she is associated (e.g. work place, and previous affiliations), related place names (e.g. place of residence, and previous attachments) and also related email ids. These facts represent the identity of that particular individual and the overlapping of different webpages through these entities act as a driving factor for a web-user to unify them referring to the same individual. We assume that the more the webpages overlap through these entities, the higher is their chance of referring to the same person.

- *Structure Overlapping*: Hyperlinks are commonly used in webpages for easy navigation to detailed reference of a fact or entity, and the inter-connection of these pages through them make up the Web structure. Generally, different webpages referring to the same person are very likely to have hyperlinks that redirect to another common webpage, regardless of its anchor text. Based on this aspect, we assume the more the webpages overlap through these hyperlinks, the higher is their chance of referring to the same person.

- *Local Context Overlapping*: The name of a person in a webpage generally accompany with it a set of closely related key words as its neighbor in the text. For example, on searching for the name *Albert Einstein* on the Web, the following sentence is found in a page.

  "...Albert Einstein (14 March 1879 – 18 April 1955) was an agnostic Jewish German-Swiss-Austrian-American physicist who is widely regarded as one of the most influential scientists of all time...."

  The neighboring words of the name present a local context and generates some sense of background knowledge about this person. If we observe the neighboring words of *Albert Einstein*, we can find that these words will frequently be found whenever the page relates to this *scientist* Einstein. We capture this aspect using a similarity measure of local context using the neighboring terms.

Most of the time, all the words in a webpage are not relevant and significant to represent the identity of the person to whom the pages refer. For example, the stop words rarely contribute to any worthy information. Rather only few important terms, that are very often called *keywords*, or *keyphrases* if they comprise multiple words, bring forth the core concept of the page, which can be collected by applying a suitable keyword extraction technique [2]. But in our case, the problem is concerned with names of a person. However, to deal with person names, entities like name of other persons related to the person for whom alias mining is being done, places to which the person is related, organizations with which the person is associated, etc., serve as better candidates to be considered instead of keywords, as suggested in previous works [11]. Therefore we finally arrived at named entities appearing on webpages to consider as the most significant factor to identify the identity of each individual namesake. Subsections 3.3 and 3.4 describe in detail the namesake profile disambiguation process.

3.3. Entity Extraction

Upon analyzing several personal homepages and going through the literature, we found five basic types of entities that play a key role in identifying an individual and aid in distinguishing between different namesakes, which are *person names*, *place names*, *organization names*, *email ids* and *hyperlinks*. Out of them, first four entities are used to capture the content-overlapping, whereas the last one is used to capture the structure-overlapping. After the target webpages for the person p are collected and pre-processed following the method in section 3.1, all the entities falling in any of the above mentioned five categories are extracted from their contents. We use Stanford NER[4] to identify and extract the first three entity types,

[3] Google search engine performs a search for partial match by default and perfect match when query text is embedded within double qoutes.

[4] `http://nlp.stanford.edu/software/CRF-NER.shtml`

and a customized HTML Parser[5] to identify and extract email ids and URLs.

3.4. Namesake Identification

Based on the extracted set of entities, this section presents a graph-based method to identify the different namesakes in the collected data. A preliminary version of this task [5] has been published, in which our focus was limited to identifying the different namesakes and generating their profile summaries from a set of collected webpages. The methodology followed in that work primarily consisted of five steps, namely, webpage retrieval and content extraction, entity extraction, graph generation, Markov clustering, and profile summary generation. In contrast, the current work focusses broadly on the namesake alias mining with extensive experimental evaluations.

Figure 2 shows the control and the data flows using solid and dotted line arrows respectively between its two sub-tasks, *graph generation* and *Markov clustering*. Further detail about their internal working is described in the following subsections.

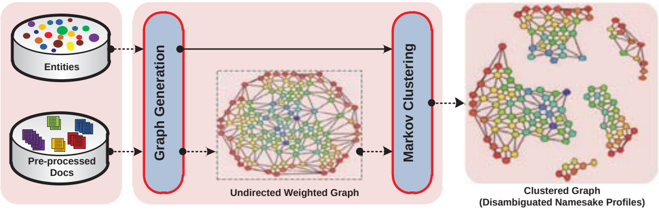


Figure 2: Namesake identification process

3.4.1. Graph Generation

Definition 1: (**Graph**) A graph is an ordered pair $G = (N, L)$, where N is the set of nodes and L is the set of links connecting the nodes in the graph. It can also be said as a simplified pictorial representation of a set of relationships L existing in between a set of objects N, in which the objects are represented in the form of nodes and the relationships between the pair of objects are represented in the form of links between the participating nodes.

Definition 2: (**Weighted Graph**) A weighted graph is an ordered 3-tuple $G = (N, L, W)$, where N is the set of nodes, L is the set of links connecting the nodes in the graph, and $W = \{w : L \rightarrow \mathbb{R}\}$ is the set of weights associated with each link. It can also be said as a simplified pictorial representation of a set of relationships L existing in between a set of objects N, where each relationship is associated with some degree of strength called weight $w \in W$. The objects are represented in the form of nodes and the relationships between the pair of objects are represented in the form of weighted-links between the participating nodes.

Definition 3: (**Graph Data Structure**) A graph data structure is used to store a graph digitally and apply computational operations on it. It comprises a finite set of nodes N to represent the objects in it, and their unordered pairs (or an ordered pair in case of a directed graph) accompanied with weights $w \in W$ (in case of a weighted graph) to represent the links (or relationships) L between them.

Definition 4: (**Adjacency Matrix**) The adjacency matrix A of a graph G is a $n \times n$ zero-one (or any value for a weighted graph) matrix with 1 (the weight value in case of weighted graph) as its $(i, j)^{th}$ entry when n_i and n_j are adjacent, and 0 as its $(i, j)^{th}$ entry when n_i and n_j are not adjacent. In other words, if its adjacency matrix is $A = \left[a_{ij}\right]$, then the value of a_{ij} is as given in equation 1, where w_{ij} is the link weight for a weighted graph and 1 for an un-weighted graph.

$$a_{ij} \leftarrow \begin{cases} w_{ij}, & \text{if } (n_i, n_j) \text{ is an edge of } G \\ \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

The proposed approach models the extracted entities in the form of a graph. As per the definitions presented above, a weighted graph is composed of sets of objects, relationships and relationship strengths. We consider all the extracted entities belonging to the five entity types and the webpages as objects, and the $\langle$*entity-entity*$\rangle$, $\langle$*entity-webpage*$\rangle$, and $\langle$*webpage-webpage*$\rangle$ type relationships as relationships between the objects. These are modeled in the form an undirected-weighted graph with a total of n nodes corresponding to the set of objects.The graph data structure used to store and apply computations on it digitally is its adjacency matrix. Algorithm 1 presents the graph generation algorithm where we can see that the input is a set of retrieved webpages and generated output is an undirected-weighted document graph modeled from the given input. Regardless the number of webpages containing an entity, a single node is added to represent all the appearances in those webpages. For each hyperlinked URL entity, one extra node is added to the graph that corresponds to *seed URL* of the actual hyperlinked URL and again only unique seed URLs are added. This is done because of the reason that it's very unlikely to match the exact full hyperlinked URL of two different webpages even if they belong to the same person, but at the same time it's very likely that the two pages of the same person are hyperlinked to two different webpages on the same website (seed URL). For example, although the hyperlinks `http://www.w3.org/People/Berners-Lee/\#Bio` and `http://www.w3.org/People/Berners-Lee/Longer.html` are not exactly the same but they are highly related. The first URL directs to a brief biography of the famous scientist *Tim Berners Lee*, whereas the latter one directs to a detailed biography. So, a unique seed URL `http://www.w3.org` is added into the graph to act as a linking node for them and highlight their relatedness. Seed URLs are also added for each of the email ids. For example, for an email id `timbl@w3.org`, a seed URL is added as a node to the graph corresponding to `http://www.w3.org`, subject to its uniqueness in the graph. It is used as a linking node to highlight the relatedness between any two different entities of the types, email id and URL ($\langle$*emailid-emailid*$\rangle$ or $\langle$*emailid-URL*$\rangle$ or

[5] `http://htmlparser.sourceforge.net/`

Algorithm 1: Graph generation(Set of webpages D)

```
nodes $N \leftarrow \Phi$, links $L \leftarrow \Phi$, weights $W \leftarrow \Phi$;
forall the $d_i \in D$ do
    $n \leftarrow createNode(d_i)$;
    $N \leftarrow N \cup \{n\}$;
    $E \leftarrow extractEntities(d_i)$;
    forall the $e_j \in E$ do
        $n \leftarrow createNode(e_j)$;
        $N \leftarrow N \cup \{n\}$;
        if $e_j$is hyperlink then
            $n \leftarrow createNode(getSeedURL(e_j))$;
            $N \leftarrow N \cup \{n\}$;
forall the node pairs $(n_i, n_j) \in N \times N$ do
    if both are pages then
        $L \leftarrow L \cup \{(n_i, n_j)\}$;
        $W \leftarrow W \cup \{\omega(n_i, n_j)\}$ ;  // using equation 12
        ;
    else if one is page and other is entity then
        $L \leftarrow L \cup \{(n_i, n_j)\}$;
        $W \leftarrow W \cup \{\omega(n_i, n_j)\}$ ;  // using equations 3
        and 5
        ;
    else if (both are entities) $\wedge$
    (co-occurrence$(n_i, n_j)$ = sentencelevel) then
        $L \leftarrow L \cup \{(n_i, n_j)\}$;
        $W \leftarrow W \cup \{\omega(n_i, n_j)\}$ ;   // using equation 2
        ;
    else if (one is seed URL and other is email-id or URL)
    $\wedge$ (isContained$(n_i, n_j)$ = true) then
        $L \leftarrow L \cup \{(n_i, n_j)\}$;
        $W \leftarrow W \cup \{\omega(n_i, n_j)\}$ ;  // using equations 6
        and 7
        ;
$G \leftarrow (N, L, W)$;
return $G$;
```

⟨*URL-URL*⟩). The nodes are connected among them by links to incorporate relationships, which fall in either of the three categories that we capture - ⟨*entity-entity*⟩, ⟨*entity-webpage*⟩, and ⟨*webpage-webpage*⟩. The degree of strength of the relationships are also computed to assign weights to them based on some formulations, described in the following.

- For each pair of entities e_i and e_j that has co-occurrence at the sentence-level, a link between them is established and its weight $\omega\left(e_i, e_j\right)$ is calculated using equation 2, which is defined as normalization of the total no. of co-occurrences of e_i and e_j at sentence-level in the whole set of documents D.

$$\omega\left(e_i, e_j\right) = \frac{\sum\limits_{D} freq\left(e_i, e_j\right)}{\max\limits_{p,q,p \neq q} \left\{ \sum\limits_{D} freq\left(e_p, e_q\right) \right\}} \tag{2}$$

- For each entity e_i extracted from a document d_j, a link between e_i and d_j is established and its weight is calculated using equation 3, which is defined as ratio of the frequency count of e_i in d_j to the maximum frequency count of the entities belonging to the class of e_i in d_j.

$$\omega\left(e_i, d_j\right) = \frac{freq\left(e_i, d_j\right)}{\max\limits_{type(e_i)} \left\{ freq\left(e_k, d_j\right) \right\}} \tag{3}$$

- For each seed URL s_i, standing for full hyperlinked URLs and/or e-mail ids, present in a document d_j, a link between s_i and d_j is established and its weight is calculated using equation 4, which is defined as ratio of the frequency count of s_i in d_j to the maximum frequency count of the entities of this category in d_j.

$$\omega\left(s_i, d_j\right) = \frac{freq\left(s_i, d_j\right)}{\max\limits_{type(s_i)} \left\{ freq\left(s_k, d_j\right) \right\}} \tag{4}$$

This measure is used to highlight the relationship of a webpage with seed URLs of the redirect hyperlinks in it, and it is intended to assign a higher weight to those seed URLs which are more relevant to the corresponding webpage. However we observed that (4) reflects some level of biasness in some cases. For example, it is very common to find hyperlinks redirecting to Wikipedia webpages for entirely different reasons, which most of the time causes the frequency of the seed URL `http://www.wikipedia.org` to be the highest of all. As a result it shows a strong relationship between the webpage and Wikipedia, and a weak relationship with any other seed, say `http://www.fbi.gov`, with fewer hyperlinks redirecting to this website. There are many other websites like Wikipedia to show this kind of behavior. Therefore, to control this biasness we enhanced (4) by multiplying a *biasness control* factor into it, as shown in equation 5, where h is the no. of unique hyperlinked URLs associated with s_i. It causes the widely varied URLs to decrease its value at a faster rate than those which are less varied comparatively.

$$\omega\left(s_i, d_j\right) = \frac{freq\left(s_i, d_j\right)}{\max\limits_{p} \left\{ freq\left(s_p, d_j\right) \right\}} \times \frac{1}{\log_2 h + 1} \tag{5}$$

- Each seed URL is actually related to the hyperlinked full URLs from which it has been derived, and this relationship can be incorporated into the graph by establishing links between the hyperlinked full URLs and the seed URLs derived from them. Therefore, for each full URL u_i, a link is established between u_i and the seed URL s_j derived from u_i, and its weight is assigned as 1.

$$\omega\left(u_i, s_j\right) = 1 \tag{6}$$

- Email addresses play a key role in identifying an individual. Generally people have their email ids in the domain of their organization website to which they belong or any how are related. To incorporate this relation, for each email id l_i, a link is established between l_i and the seed URL s_j derived from l_i, and its weight is assigned as 1.

$$\omega\left(l_i, s_j\right) = 1 \tag{7}$$

- We also consider the relationship between each pair of webpages that is defined by the level of similarity between them. For this, considering the set of m retrieved webpages as the corpus, $D = \{d_1, d_2, ..., d_m\}$, and each of the extracted entities from them as the set of n distinct terms, $T = \{t_1, t_2, ..., t_n\}$ we construct a *term-document matrix*, $\Omega_{n\times m}$, where the value of the element, $\Omega(i, j)$, is the $TF\text{-}IDF$ value of the term, t_i, in document, d_j. Nowadays researchers apply several different kind of formulations to calculate the value of $TF\text{-}IDF$ depending on the nature of the problem. One of our previous works [2] show some commonly used formulations and their effectiveness for keyphrase ranking. Kalashnikov *et al.* worked for namesake disambiguation in [23] in which they used the formula for TF as shown in equation 8. It makes it to range in between 0.5 and 1. In our experiments we found that although this version performs equally to that in [39] for ranking prominent terms by this value, but behave in an unexpected manner when used for measuring cosine similarity. The reason behind this is the presence of terms with zero frequency in this case. When a term is having zero frequency in a document, as will be with majority of terms, its TF value is 0.5 and if the IDF value is high, the combined value becomes fairly high, even higher than those with non-zero frequencies but comparatively low IDF values. When this value is applied to compute cosine similarity between two documents not having any common term, a sufficiently high value is returned, very often even higher than those with common terms in them. The contribution of the TF part in $TF\text{-}IDF$ is excessively minimized as compared to IDF which is the responsible factor for this kind of behavior. Hence, for our approach we found equation 9 to be performing well.

$$TF\left(t_i, d_j\right) = \frac{1}{2} + \frac{freq\left(t_i, d_j\right)}{2 \times \max\limits_{k}\left\{freq\left(t_k, d_j\right)\right\}} \tag{8}$$

$$TF\text{-}IDF\left(t_i, d_j\right) = \frac{freq\left(t_i, d_j\right)}{\max\limits_{k}\left\{freq\left(t_k, d_j\right)\right\}} \times \log_2 \frac{m}{freq\left(t_i\right)} \tag{9}$$

Using Ω, *content similarity* C is measured as the cosine similarity between each pair of webpage nodes to act as the weight of that link as shown in equations 10 and 11.

$$C\left(d_i, d_j\right) = \frac{\sum\limits_{l=1}^{n} \Omega(l, i) \times \Omega(l, j)}{\sqrt{\sum\limits_{l=1}^{n} \Omega(l, i)^2}\sqrt{\sum\limits_{l=1}^{n} \Omega(l, j)^2}} \tag{10}$$

$$\omega\left(d_i, d_j\right) = C\left(d_i, d_j\right) \tag{11}$$

However, later on we observed another important thing. The entities extracted from webpages are not the only factors to stand on behalf of them for similarity measurement, rather there exist other informative fundamental terms that present a key concept. These are the general English words that present a background picture of the person about whom it is described therein. For example, if the name found in a webpage is of a person who is a professor or scientist then some of the common words in this page will be as *teaching*, *research*, *students*, etc., where as if the person is a tennis player these words could be as *wimbledon*, *match*, *won*, *lost*, etc. Although most of these words follow a PoS tag pattern of *nouns* and *verbs* but are not limited to it. Moreover, the words appearing closer to the name on either of its sides (before or after) on a page are more likely to be relevant to that person. Considering all these factors, we designed a similarity measure highlighting the *local context* of that name to improve the current measure of similarity between different webpages. For each webpage d_i, we generate a set of local context words X_i. For this, the content text is tokenized into several chunks where chunk boundaries are determined heuristically by various punctuation marks. Setting a window size of w_c for each occurrence of the person name in a chunk from d_i, w_c words or unigrams adjacent to the name occurring both before and after it are added to the set X_i, if they exist and are not a stopword according to our list of 583 stopwords. From these sets of context words for all the webpages in D a unified set, $U = \{u_1, u_2, ..., u_n\}$, is generated containing all the unique context words to represent the complete *context* of that name. Thereafter, a *context-document matrix* $\Gamma_{n\times m}$ is constructed where value of the element $\Gamma(i, j)$ is frequency of the context word u_i in the local context set X_j. At last the *local context similarity* $L\left(d_i, d_j\right)$ is measured as the cosine similarity between each pair of documents, d_i and d_j, using the matrix $\Gamma_{n\times m}$ in the same way as content similarity. And the redefined weight for the link between each pair of webpages is calculated using equation 12, where, $0 \leq \mu \leq 1$ is a balance factor constant.

$$\omega\left(d_i, d_j\right) = \mu \times C\left(d_i, d_j\right) + (1 - \mu) \times L\left(d_i, d_j\right) \tag{12}$$

3.4.2. Markov Clustering

The state-of-the-art solutions for this problem, as discussed in section 2, have found graph-based clustering techniques as the most successful because of its nature [23, 30]. As discussed in section 3.4.1, we generated an undirected-weighted graph from the complete set of retrieved webpages showing each of the entities as nodes and relationships among them as weighted links. As our next step, we apply *Markov clustering* (MCL) [44, 49], which transforms G into a directed graph G_i with several weakly connected components. These components form a set of clusters, each of which represent an individual and the webpages in it refer to that person.

Definition 5: (**Markov Matrix**) A *Markov matrix* also called as *row stochastic matrix* is a matrix satisfying the condition that each matrix element $m_{(p,q)} \geq 0$, and each row adds up to 1.

Definition 6: (**Markov Chain**) A *Markov chain* is a discrete-time stochastic process over a set of states satisfying the Markov property, which means that the probability distribution of X_i, over the states at time i, depends only on the probability distribution of X_{i-1}, as shown in equation 13.

$$P\left(X_i = p|X_0, X_1, \cdots, X_{i-1}\right) = P\left(X_i = p|X_{i-1}\right) \tag{13}$$

MCL is a graph clustering technique based on simulating a random walk on a weighted graph. It considers the transition from one node to another within a cluster as much more likely than those in different clusters, taking into account the weight of those links. This algorithm accepts the adjacency matrix $A_{n \times n}$[6] of graph G as an input and starts working by adding loops or self-edges to A by applying $a_{pp} \leftarrow 1$, if they do not exist, and converting this matrix to a *Markov matrix* $M_{n \times n}$. M acts as a transition matrix for a Markov chain or a Markov random walk on G. The rest of the algorithm is an iterative method interleaving matrix expansion by multiplying with itself and inflation steps using equation 14 that keeps on iterating until the transition matrix M_i converges. Most of the clustering algorithms need to provide the required number of clusters k as an input parameter, which is often difficult to determine beforehand. Unlike them, MCL is free from this limitation, instead it uses the inflation parameter $r > 1$ to decide the value of k. It determines the level of disambiguation that has to be applied and the exact number of clusters can't be guaranteed in advance. Higher values of r lead to more and smaller clusters in size, whereas smaller values lead to fewer and larger clusters. The difference between the successive matrices is calculated in terms of the *Frobenius norm* shown in equation 15. The generated matrix M_i after its convergence in the i^{th} iteration, results to a directed graph with weakly connected components in it. The nodes having values greater than zero in the diagonal, i.e., $m_{pp_i} > 0$, are called as *attractors* of the corresponding cluster. All other nodes having a link with the attractor are attracted towards it and are included in that cluster. If a node is attracted towards multiple attractors then there is an overlapping of those clusters to which these multiple attractors correspond.

$$\xi\left(M, r\right) = \left\{ \frac{\left(m_{pq}\right)^r}{\sum_{a=1}^{n} \left(m_{pa}\right)^r} \right\}_{p,q=1}^{n} \tag{14}$$

$$\|M_i - M_{i-1}\|_F = \sqrt{\sum_{p=1}^{n} \sum_{q=1}^{n} \left(m_{pq_i} - m_{pq_{i-1}}\right)^2} \tag{15}$$

Thus, the clustering algorithm splits up the graph into several clusters comprising webpage nodes as well as entity nodes. Each of these clusters is mapped to a corresponding namesake individual with the common name, and any kind of information regarding a specific individual can be mined from the corresponding cluster to generate a profile. In this paper, we apply an alias mining technique to extract aliases corresponding to each namesake individual.

3.5. Alias Identification

In this module, the set of webpages in each of the separated clusters are mined to identify alias names for each namesake. As shown in figure 3, it comprises three sub-tasks, namely, *candidate alias identification*, *feasibility analysis* and *ranking*. Further details about each of them are provided below.

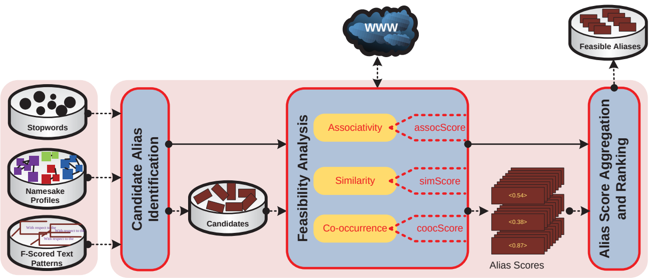


Figure 3: Alias identification process

3.5.1. Candidate Alias Identification

Generally aliases are used on the Web with different motives. Sometimes, the author of the page uses an alias name instead of real name just for being easy to spell (e.g. *Alby* instead of *Albert Einstien*), or focusing on specific activity or interest in relation to something of importance (e.g. *the father of modern science* instead of *Albert Einstien*), or hiding the identity from rest of the public and conveying the message to those intended for (e.g. *Hajj* is an alias name of "*usama bin laden*" as per *FBI* records, but until it was known to them it might have been used to indicate him by his network partners). At other times may be unintentionally or intentionally the author introduces an alias name of the person to relate it with the real name. For example, in the following sentence found on a webpage, the author introduces the alias name of Albert Einstien.

...Albert Einstein also known as the father of modern science invented made many discoveries. His work is very important for society ...

[6] As G is an undirected graph, so its adjacency matrix A will be symmetric in nature

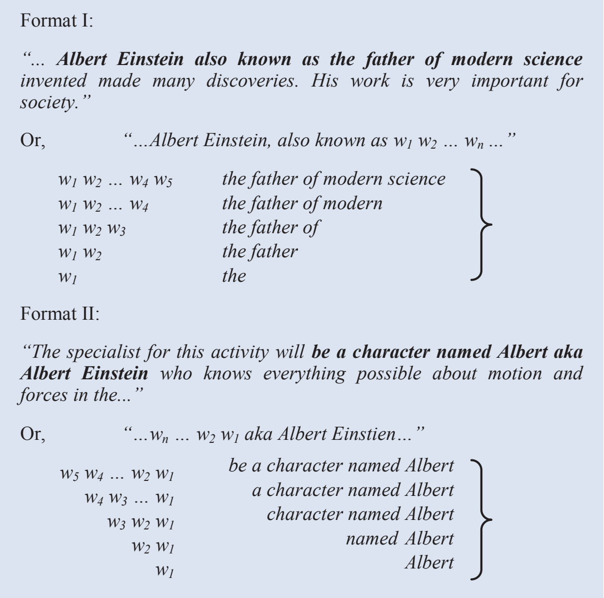
Table 2: F-scored Patterns [10]

Pattern No.	Pattern-Based Querries	F-score
p_1	SearchQuery(*, *aka, realName*)	0.335
p_2	SearchQuery(*realName, aka,* *)	0.322
p_3	SearchQuery(*realName, better known as,* *)	0.310
p_4	SearchQuery(*realName, alias,* *)	0.286
p_5	SearchQuery(*realName, also known as,* *)	0.281
p_6	SearchQuery(*, *nee, realName*)	0.225
p_7	SearchQuery(*realName, nickname,* *)	0.224
p_8	SearchQuery(*, *whose real name is, realName*)	0.205
p_9	SearchQuery(*realName, aka the,* *)	0.187
p_{10}	SearchQuery(*, *was born, realName*)	0.153

In [11], Bollegala *et al.* worked on the same problem. Through experiments in their research they discovered the top ranking text patterns that are generally used in webpages to associate alias names with the real name, and also calculated *F-scores* of each text pattern on the Web. Table 2 presents 10 top F-scored text patterns out of those extracted by them. In an earlier work they had used just 25 top F-scoring patterns [10], but later on they experimented by raising the number of patterns to find its effect on overall results. It was found that raising it up to some extent improves the overall result because of the substantial increase in true positives and decrease in false negatives. Therefore in their latest work [11], they used top 200 text patterns to identify the candidate aliases. For the candidate extraction task, we follow the same approach and from each cluster of webpages we search for the matching text patterns. All the matching sentences are collected and after some ad hoc cleaning technique to smoothen the noise in the unstructured text, they are divided into record-size tokens. The token boundaries are determined heuristically by various punctuation marks. Thereafter, each token is subjected to a windowing technique for text chunk generation. It can be observed from table 2 that some patterns (like p_2, p_3, p_4, p_5, p_7 and p_9) have wildcard character (*) as a right-most constituent, whereas others have it as a left-most constituent. Hence, depending on the pattern associated with a retrieved token, text chunk is generated either from right-side texts or from left-side texts. Moreover, the chunks are extracted differently, from the right in first case and from the left in the latter one. Figure 4 presents an exemplar text token and retrieved chunks for both cases. The chunks either with a complete match in the list of stopwords or those begining or ending with a stopword are filtered out, and the remaining are accepted as candidate aliases with its triplet as $\langle candidate, frequency, pattern \rangle$, including the text pattern in which the candidate appeared and its frequency count in the corresponding pattern.

3.5.2. Feasibility Analysis

The candidate aliases extracted in section 3.5.1 are those that somehow possess some key property in them to get marked as aliases, for which they can also be called as *"to be aliases"*. However due to the unstructured nature of the Web, we may also be misled by restricting ourselves to the previous step. Therefore we employ some additional statistical measures to determine the candidacy of an alias and analyze their feasibility. Our statistical mechanism captures three diverse and salient properties of an alias for feasibility analysis which are as given below.



Format I:

*"... **Albert Einstein also known as the father of modern science** invented made many discoveries. His work is very important for society."*

Or, *"...Albert Einstein, also known as w_1 w_2 ... w_n ..."*

w_1 w_2 ... w_4 w_5 the father of modern science
w_1 w_2 ... w_4 the father of modern
w_1 w_2 w_3 the father of
w_1 w_2 the father
w_1 the

Format II:

*"The specialist for this activity will **be a character named Albert aka Albert Einstein** who knows everything possible about motion and forces in the..."*

Or, *"...w_n ... w_2 w_1 aka Albert Einstien..."*

w_5 w_4 ... w_2 w_1 be a character named Albert
w_4 w_3 ... w_1 a character named Albert
w_3 w_2 w_1 character named Albert
w_2 w_1 named Albert
w_1 Albert

Figure 4: An example showing the windowing technique to extract text chunks where the window is set to 5

Associativity: When a candidate alias appears along with its real name in the same sentence, the relevance of the candidate is determined by its association with the real name, and it depends very much on the text pattern connecting them. In other words, the more reliable a text pattern connecting the candidate alias to real name, the higher the potentiality of this candidate to be an alias.

Similarity: In the webpages, where candidate alias has been used in its own without referencing real name, the context of its surroundings remains the only key factor to capture its identity. The more similar the context of a candidate with its real name, the higher the potentiality of this candidate to be an alias.

Co-occurrence: Generally, when an author creates a webpage to introduce a person by using his or her alias name, the author may also use the real name somewhere else in that page, i.e., the co-occurrence of both the names on a webpage improves the chances of a candidate to be an alias. Therefore, the more the number of co-occurring webpages, the higher the potentiality of this candidate to be an alias.

The above three aspects are captured by three different scores, *assocScore*, *simScore* and *coocScore* respectively as defined below.

Association Score Calculation: Each of the candidate aliases is assigned an association score that determines its associativity with the text pattern with which it appeared in the webpage. Since a candidate alias a may exist with multiple patterns, an aggregated association score $assocScore(a)$ is calculated using equation 16 where i varies over the number of text patterns associated with a, $F\text{-}score(t_i)$ denotes the F-score value of an associated pattern t_i and $freq(a_i)$ denotes the number of times

a occurs in association with t_i.

$$assocScore(a) = \sum_i (F\text{-}Score(t_i) \times freq(t_i)) \quad (16)$$

Similarity Score Calculation: Generally when name of a person is used somewhere on the Web, the accompanying terms describe something about that person and the person is somehow related to those words. Despite that whether it is the real name or alias name, the accompanied neighboring words present the context or setting of that entity. So, for each candidate alias and the real name, we generate context sets and use them to find context similarity between each alias name and the real name. The contexts of both alias and real name are represented in terms of the vector space model and cosine similarity is applied between them. Higher values indicate better similarities and in turn more resemblance to real name as compared to those with lower values.

In section 3.4.1, for each webpage document d_i, a set of local context words X_i is generated. To get the context of real name of a namesake S_{rn}, a set-theoretic union operation is applied between the context sets X_i for all documents d_i in the corresponding cluster, i.e., $S_{rn} = \bigcup_i X_i$. To generate the context set of its alias name a, a search query is passed in corresponding cluster of webpages for the candidate alias. The matching webpages are tokenized, cleaned, and w_c adjacent unigrams from left of the query string and another w_c unigrams from its right are collected after filtering out the stopwords. Thus the corresponding context set S_a is generated following the same process as explained earlier. Similarity between their contexts is measured by transforming the context sets, S_{rn} and S_a, to their vector-space model representations, V_{rn} and V_a, and computing the cosine similarity between them as shown in equation 17.

$$simScore(a) = \frac{\sum_{i=0}^{n} V_{rn}(i) \times V_a(i)}{\sqrt{\sum_{i=0}^{n} (V_{rn}(i))^2 \times \sum_{i=0}^{n} (V_a(i))^2}} \quad (17)$$

Co-occurrence Score Calculation: The *co-occurrence score* measure is applied to boost up the candidate aliases which are more frequent in webpages along with their real names, as compared to those with less frequent co-occurrences. Although it is also a type of associativity measure between them, the prime focus here is not their association, because there doesn't exist any specific pattern or link to establish a strong relationship. Despite their placement fashion or their lexical and syntactic structure, it just depends on the count of their co-occurrences on the same page. We performed experiments with several different types of score values, but finally arrived at *Dice Coefficient* producing the best results. This measure is also used by Bollegala *et al.* [11] as one among several association measures. The co-occurrence score value for each candidate alias a and its real name rn is calculated using (18). It interacts with the WWW to find out the number of hits for a name or alias on the Web.

$$coocScore(a) = \frac{2 \times hits(rn\ AND\ a)}{hits(rn) + hits(a)} \quad (18)$$

3.5.3. Ranking

In the previous sub-task, feasibility of candidates is analyzed in terms of three statistical measures. Unifying all the three scores to a single ranking value, this step determines the comparative feasibility and prominence of the candidate aliases. They are unified to get an aggregated value called *aliasness* of a candidate, formulated in equation 19, where AS, SS, and CS are assocScore, simScore, and coocScore respectively, and $\Lambda = \{\lambda_1, \lambda_2, \lambda_3\}$ is a set of three constant values such that $\sum_{\lambda_k \in \Lambda} \lambda_k = 1$. These constants act as control on each pair of scores and their values maintain the balance between them.

$$\begin{aligned} Aliasness(a) &= \lambda_1 \times (AS(a) \times SS(a)) \\ &\quad + \lambda_2 \times (AS(a) \times CS(a)) \\ &\quad + \lambda_3 \times (SS(a) \times CS(a)) \end{aligned} \quad (19)$$

Finally based on this aggregated value of *aliasness*, candidate aliases are ranked. r top aliases are declared as feasible and most promising aliases for the corresponding namesake, that are being used on the Web as an alternate to a real name. The task of *alias identification* is performed repeatedly for each cluster generated in subsection 3.4.2 to mine aliases of each of the corresponding namesakes.

4. Experiments

As we saw in section 3, the proposed approach comprises two core tasks, namesake profile disambiguation covered in subsections 3.3 and 3.4, and alias mining covered in subsection 3.5. We conducted several experiments to bring forth the effects of different individual components and establish their efficacy. Finally they are integrated altogether to evaluate the overall results.

4.1. Datasets

The experiments are conducted on three different real datasets to evaluate all the different components. The namesake profile disambiguation approach is evaluated on the datasets of Bekkerman and McCallum [9] and the WePS-2 test dataset [7]. The Bekkerman dataset consists of text files created by filtering the html tags from webpages for 12 different person names, which are the names of SRI employees and professors from different universities. Table 3 shows its statistics, in which the webpages for each name have been manually annotated into multiple categories based on their content. Whereas the WePS-2 test dataset shown in table 4 consists of 30 person names, with 10 out of them collected from Wikipedia, another 10 from ACL'06 and the remaining 10 from US Census. To evaluate the complete methodology of namesake alias mining, we used the dataset of Bollegala *et al.* [11]. It consists of 50 English person names, 50 English place names of US and 100 Japanese person

Table 3: Statistics of Bekkerman dataset

Person name	Position	Pages(#)	Categories(#)	Relevant pages(#)
Adam Cheyer	SRI Manag	97	2	96
William Cohen	CMU Prof	88	10	6
Steve Hardt	SRI Eng	81	6	64
David Israel	SRI Manag	92	19	20
Leslie Pack Kaelbling	MIT Prof	89	2	88
Bill Mark	SRI Manag	94	8	11
Andrew McCallum	UMass Prof	94	16	54
Tom Mitchell	CMU Prof	92	37	15
David Mulford	Stanford Undergrad	94	13	1
Andrew Ng	Stanf Prof	87	29	32
Fernando Pereira	UPenn Prof	88	19	32
Lynn Voss	SRI Eng	89	26	1
Overall		1085	187	420

names. We consider only the English person names for our experiments, each of which have their gold standard aliases along with them.

4.2. Evaluation Metrics

We use different standard metrics in the experiments to evaluate the quality of results and compare with existing works on related problems. Purity F-measure F_P and B-Cubed F-measure F_B are two standard metrics commonly used to evaluate cluster sets of items in general and namesake disambiguation in particular by comparing with the gold standard cluster set, and therefore we use them to evaluate the namesake disambiguation section.

F_P at $\alpha = 0.5$ (or $F_{\alpha=0.5}$) is defined as the harmonic mean of purity and inverse purity defined in equations 20 and 21 respectively, . For a system producing C clusters for a dataset with $|W|$ elements (webpages) manually annotated into L set of categories, purity and inverse purity are calculated by using equations 20 and 21 respectively.

$$Purity = \sum_{C_i \in C} \frac{|C_i|}{|W|} \times \max_{L_j \in L} \frac{\left|C_i \bigcap L_j\right|}{|C_i|} \tag{20}$$

$$InversePurity = \sum_{L_i \in L} \frac{|L_i|}{|W|} \times \max_{C_j \in C} \frac{\left|C_j \bigcap L_i\right|}{|L_i|} \tag{21}$$

Introduced in WePS-2, $F_{B\text{-}Cubed}$ or F_B is defined as the harmonic mean of B-Cubed precision and B-Cubed recall. For each element (or webpage) i, precision and recall values are computed individually using equations 22 and 23 respectively, and the overall B-Cubed precision and B-Cubed recall are computed as mean of the individual respective values.

$$Precision_i = \frac{C_i \bigcap L_i}{C_i} \tag{22}$$

$$Recall_i = \frac{C_i \bigcap L_i}{L_i} \tag{23}$$

To evaluate ranked items, there are the two standard metrics called mean reciprocal rank (MRR) and average precision (AP), which compare the experimentally identified rank with their gold standard rank. We use these metrics to evaluate the alias identification section and the overall approach.

Table 4: Statistics of WePS 2 test dataset

Sl No.	Person Name	Categories(#)	Pages(#)	Discarded(#)
		Wikipedia Names		
1	Bertram Brooker	1	128	30
2	Cavid Tua	1	134	36
3	Franz Masereel	3	126	26
4	Herb Ritts	2	127	31
5	James Patterson	4	133	33
6	Jason Hart	22	130	38
7	Louis Lowe	24	100	25
8	Mike Robertson	39	123	35
9	Nicholas Maw	1	135	36
10	Tom Linton	10	135	41
		ACL'08 Names		
1	Benjamin Snyder	28	95	40
2	Cheng Niu	7	100	7
3	David Weir	26	128	33
4	Emily Bender	19	120	31
5	Gideon Mann	2	95	6
6	Hao Zhang	24	100	13
7	Hoi Fang	21	90	28
8	Ivan Titov	5	101	28
9	Mirella Lapata	2	91	1
10	Tamer Elsayed	8	101	18
		Census Names		
1	Amanda Lentz	20	121	46
2	Helen Thomas	3	127	27
3	Janelle Lee	34	93	37
4	Jonathan Shaw	26	126	46
5	Judith Schwartz	30	124	39
6	Otis Lee	26	118	40
7	Rita Fisher	24	109	13
8	Sharon Cummings	30	113	29
9	Susan Jones	56	110	30
10	Theodore Smith	54	111	43

MRR focuses mainly on the rank of extracted aliases, whereas AP considers both the precision at each rank along with number of correctly identified aliases. MRR value is calculated using equation 24, where G is the set of gold standard and $rank_i$ is the rank of i^{th} alias of G in the ranked list of extracted aliases.

$$MRR = \frac{1}{|G|} \times \sum_{i=1}^{|G|} \frac{1}{rank_i} \tag{24}$$

For the extracted set of aliases A, AP is calculated using equation 25, where $Rel(i)$ is a binary function that returns 1 for a correctly identified alias and 0 otherwise, and $Pre(i)$ is calculated using equation 26.

$$AP = \frac{\sum_{i=1}^{|A|} Pre(i) \times Rel(i)}{|G|} \tag{25}$$

$$Pre(i) = \frac{\{G\} \bigcap \{A_i\}}{i} \tag{26}$$

4.3. Experimenting with the Disambiguation Approach

The first major task of the proposed approach is disambiguating namesake web-profiles. To experimentally evaluate this part, we carried out experiments on the Bekkerman and McCallum dataset [9] and the WePS-2 test dataset[7].

In the experiment on Bekkerman dataset, for each of the 12 persons we started with entity extraction. We used Stanford NER[7] to extract all the person names, place names and organization names from the text, and some self-designed ad hoc rules based on the content structure of text files embedded in HTML parser to extract the hyperlinked URLs and email ids. Using these entities an undirected weighted graph is generated following the approach mentioned in section 3.4.1. It is represented in the form of its weighted adjacency matrix. Finally, MCL is applied on this matrix with varying values of the inflation parameter, r, setting the threshold value for the *Frobenius norm*, θ, to 0.001. In the first stage of experimentation, we considered only five type of entities as nodes in the graph and local context overlapping is also not brought into consideration. However later on, the system is enhanced by including some novel features. An extra unique node for the *seed URL* is added to the graph for each hyperlinked URL and email id along with its links with the webpage, hyperlinked URL and email id. The objective is to map those hyperlinks which are not exactly the same but have the same *seed URL* thereby increasing the possibility of being related to the same subject. The weight assigning formula for these edges considers well enough the diverse nature of a website or a seed URL. The other extension is the introduction of local context overlapping. It is included as an additional similarity measure between a pair of webpage nodes. For our experiments the value of μ in equation 12 is set to 0.5. Table 5 shows the results found by us in both stages where we can see the B-Cubed F-measures F_B, and the purity F-measures F_P, at $\alpha = 0.5$ of each person individually. We considered five different values for the inflation parameter r which in itself doesn't decide the number of clusters to be generated, rather it creates a level of disambiguation or the extent to which two different nodes are to be considered as similar. Because of this nature even for the same value of r, it produces different number of clusters for each group of webpages depending on the content nature and in turn the weighted graph, which is very much realistic.

Figure 5 presents a comparison of no. of generated clusters for four different person names on varying values of r. We see that as we keep on increasing this value, the generated number of clusters increases, however the exact no. of clusters cannot be predicted by it. No. of Clusters generated for *Tom Mitchell* grows much more rapidly than *Adam Cheyer*. At the value of 1.10 it generates 5 and 1 clusters for them respectively, whereas at 1.20 these values rise to 52 and 7 making a huge difference. The reason behind this is the nature of their content. Moreover, we can see in table 5 that the best result for the individual person names do not have any direct relationship with the value of r. For *Adam Cheyer* the best F_B/F_P values, 99.0/99.0, are found at 1.10, where as for *David Israel* we find the best result at 1.17. Looking into the complete dataset we see that the highest values for F_B/F_P are concentrated nearby 1.15, with best values for three names at 1.13, another three at 1.15 and another four at 1.17. Hence, calculating the average F-measures of all the individual values we found the best result when it is set to 1.15 for both F_B as 77.2% and F_P as 81.4%. One interesting point to be noted is that the B-Cubed measure has always remained higher than that of purity. Figures 6(a) and 6(b) present the impact of the inflation parameter on the evaluation metrics, F_B and F_P. We can see the rise and fall of these values on varying the values of r. The two figures show a common trend in the variations. Whenever F_B is rising, a similar reaction is shown by F_P, and vice versa.

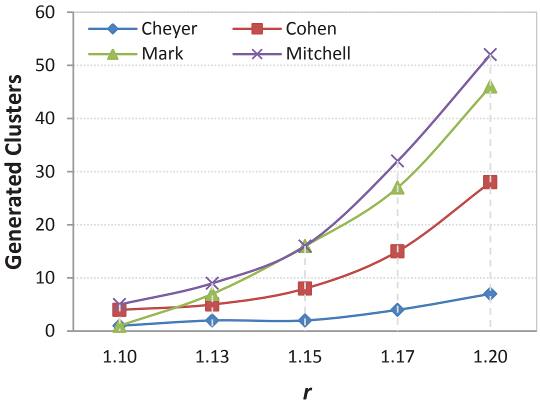
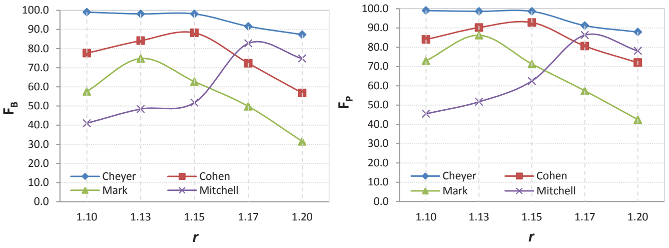


Figure 5: A comparison of no. of clusters generated for four different names



(a) Impact on F_B measure

(b) Impact on F_P measure

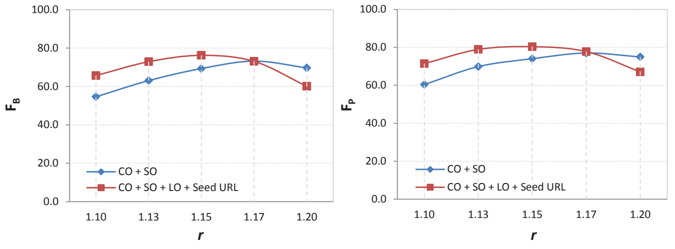
Figure 6: Impact of r on overall results

Figures 7(a) and 7(b) highlight relevance of the newly added features. The blue line shows behavior of the system (CO + SO) which considers only the *content overlapping* and *structure overlapping* with the five entity types used to generate the graph. The red line shows behavior of the system (CO + SO + LO + Seed URL) which considers the newly added *local context overlapping* feature as an additional similarity measure between the webpage node pairs in the graph and an additional node in the graph for *seed URL*. Introducing local context similarity improved the webpages' similarity measure, and the concept of seed URL increased the no. of nodes and the links between them in the graph. These links play a crucial role in connecting hyperlinks, be it a URL or an email id, that are not exactly same but are having the same seed. By connecting simi-

[7] Kalashnikov *et al.* [23] in their experiments found the quality of results of Stanford NER and GATE as comparable.

Table 5: Results on Bekkerman dataset

Person name	CO + SO					CO + SO + LO + Seed URL				
	r = 1.10 F_B/F_P	r = 1.13 F_B/F_P	r = 1.15 F_B/F_P	r = 1.17 F_B/F_P	r = 1.2 F_B/F_P	r = 1.10 F_B/F_P	r = 1.13 F_B/F_P	r = 1.15 F_B/F_P	r = 1.17 F_B/F_P	r = 1.20 F_B/F_P
Adam Cheyer	**99.0/99.0**	99.0/99.0	97.8/98.1	90.8/92.0	83.6/85.3	**99.0/99.0**	98.1/98.6	98.1/98.6	91.6/91.2	87.3/87.9
William Cohen	82.6/92.9	**84.7/93.4**	72.1/77.9	68.4/73.6	56.3/67.7	77.6/84.0	84.1/90.2	**88.2/92.8**	72.3/80.6	56.8/72.0
Steve Hardt	61.2/68.8	70.5/76.3	75.3/79.6	**83.5/85.9**	66.2/73.8	78.1/81.6	**82.4/85.7**	75.6/77.1	66.0/71.3	49.8/62.3
David Israel	43.7/55.1	54.3/63.9	60.2/67.1	66.4/70.7	**73.6/75.1**	58.8/69.4	65.3/71.0	72.4/75.8	**76.7/82.9**	63.6/70.2
Leslie Pack Kaelbling	**99.0/99.0**	99.0/99.0	97.1/97.7	92.5/93.1	88.9/91.2	**99.0/99.0**	99.0/99.0	98.4/98.5	90.7/91.1	88.4/88.7
Bill Mark	28.4/25.6	41.6/47.1	50.5/55.3	59.9/64.3	**71.6/78.7**	57.6/73.0	**75.8/86.2**	70.6/78.3	52.8/60.4	31.5/42.5
Andrew McCallum	47.2/53.3	52.0/58.6	61.8/65.4	70.8/73.1	**76.4/79.3**	55.3/60.1	68.7/71.6	**82.5/84.8**	73.9/77.1	57.8/65.2
Tom Mitchell	27.1/39.6	44.7/61.3	65.5/72.9	**81.6/87.3**	72.6/76.9	41.0/45.5	48.4/51.7	51.8/62.5	**82.7/86.3**	74.8/78.1
David Mulford	61.4/67.2	70.5/75.8	76.3/80.2	**82.4/87.8**	68.2/77.5	73.5/80.7	76.9/85.1	**84.5/88.4**	71.6/77.9	52.5/63.7
Andreq Ng	32.0/40.6	46.4/52.8	58.9/61.7	65.3/70.2	**73.5/74.1**	44.6/49.3	51.7/63.2	73.6/77.8	**78.1/81.4**	60.3/64.9
Fernando Pereira	48.8/51.2	59.1/66.3	**68.5/79.0**	63.6/69.8	44.1/53.7	65.2/68.0	**72.4/77.5**	68.8/72.3	54.7/59.2	39.9/48.6
Lynn Voss	25.0/32.6	34.9/45.1	48.2/53.5	53.4/56.7	**61.6/66.4**	38.4/47.7	53.1/67.3	62.1/69.6	**71.0/77.9**	58.6/60.8
Average	54.6/60.4	63.1/69.9	69.4/74.0	**73.2/77.0**	69.7/75.0	65.7/71.4	73.0/78.9	**77.2/81.4**	73.5/78.1	60.1/67.1



(a) Impact on F_B measure

(b) Impact on F_P measure

Figure 7: Impact of newly added features on overall results

Table 6: Result summary of namesake disambiguation

Approach	Bekkerman dataset		WePS-2 test dataset	
	F_B	F_P	F_B	F_P
CO+SO	73.2	77.0	68.5	73.9
CO+SO+LO+Seed URL	77.2	81.4	73.1	77.6

lar hyperlinks they increase the chances of webpage nodes containing those hyperlinks to converge to the same cluster during clustering. The latter system being more dense than the former behaves more abruptly in terms of no. of clusters generated. As we see in figure 5 the line showing Tom Mitchell is rising steeply even with small variations in r, generating 52 clusters at 1.20. In general, at those values of r where the no. of generated clusters are closer to the no. of manually annotated categories, the system shows better results as compared to those producing clusters either too low or too high in number. Because of this, when r is increased beyond 1.15 the performance starts degrading as large no. of clusters are generated by the latter system. Whereas in case of the former system, as the no.of generated clusters are comparatively less, the performance starts degrading after 1.17. At 1.17 both perform almost equally. Raising it further, the former system sweeps over the latter one which may seem to be strange initially. But looking into the number of clusters generated at that value of r, it can be seen that the larger no. of clusters generated by the latter system leads to comparatively poor performance at higher values. Therefore for applying this approach to a real time environment, selecting an appropriate value for r is an important task. It needs to be done by doing some experiments on exemplar datasets which will be representing the nature of their content and will enable to fix the value of r for which it produces best results on them. Thus both the newly added features are found to improve the performance substantially. Furthermore, figure 8 presents the impact of μ used in equation 12 on the overall results while experimenting with the Bekkerman dataset using the (CO + SO + LO + Seed URL) system with the value of inflation parameter r set to 1.15. At $\mu = 0$, the system becomes (SO + LO + Seed URL), whereas at $\mu = 1$ it becomes (CO + SO + Seed URL), which do not produce results as good as at $\mu = 0.5$. It shows the role of μ in the overall namesake disambiguation task.

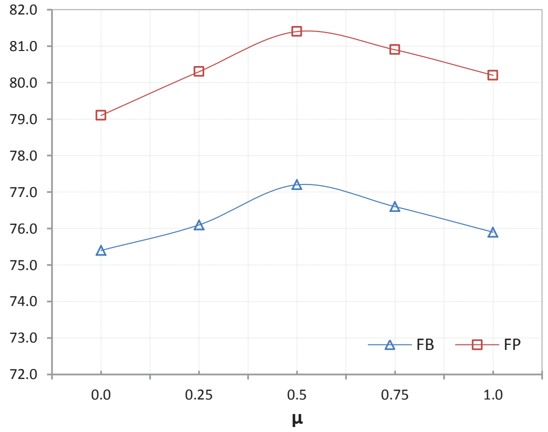


Figure 8: Impact of μ on overall results for Bekkerman dataset

In a similar way, experiments are also conducted on the WePS-2 test dataset. The overall results on both these datasets are shown in table 6. The improved system of (CO + SO + LO + Seed URL) leads to achieve the measures of F_B and F_P as

77.2% and 81.4% respectively on the Bekkerman dataset and 73.1% and 77.6% respectively on the WePS-2 test dataset. The metrics F_B and F_P, as defined and described in section 4.2, have remained among the most notable metrics for cluster evaluation. Such a good result in terms of these metrics shows the effectiveness of the proposed namesake disambiguation technique.

The Bekkerman dataset is in the form of text files generated by transforming the webpages, and doesn't contain much hyperlinks, whereas the WePS-2 test dataset exist in the form of raw webpages with large number of hyperlinks, and it makes the hyperlink *biasness control* factor, shown in equation 5, more pronounced. To evaluate the contribution of this factor we additionally conducted experiments with the system (CO + SO + LO + Seed URL) on this dataset without multiplying the biasness control factor, and found the F_B and F_P measures as 72.4% and 77.0% respectively. These values are little lower than those obtained with the factor included. Although the contribution of this factor didn't come out to be very large, the improvement in results after its inclusion definitely indicates a positive effect.

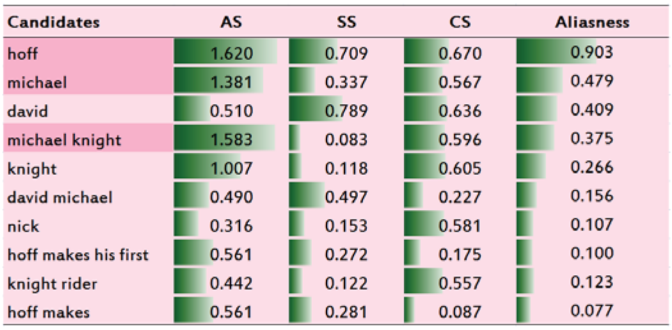
4.4. Experimenting with the overall namesake alias mining method

Our system based on Google API started the experiment with crawling a set of 500 webpages for each of the 50 names in the dataset, which are then processed for entity extraction, graph generation and clustering to disambiguate namesakes. Setting the value of the constant μ in equation 12 to 0.5 and the threshold value for the frobenius norm θ to 0.001, we generated clusters for varying values of the inflation parameter r. Table 7 presents the number of clusters generated for four test cases at five different values of r. Here again we see the same trend as in figure 5, i.e. number of clusters goes on increasing as r increases. At values 1.10 and 1.13 we get single clusters for each of the four names, then at 1.15 three clusters are generated for *Al Pacino* and two for *Teri Hatcher*, and after that this number goes up beyond the considerable limits. Upon manually reviewing all those 500 crawled webpages for each name, we found that due to the dominating popularity of these names no webpage was related to any namesake other than the person whose alias names we are looking for. Therefore, ideally we should get single clusters for each name after applying the clustering technique for disambiguation, and this ideal condition is fulfilled when r is set to 1.10 and 1.13.

Table 7: Number of clusters generated at different values of r

Real Name	**r = 1.10**	**r = 1.13**	**r = 1.15**	**r = 1.17**	**r = 1.20**
David Hasselhoff	1	1	1	2	9
Courteney Cox	1	1	1	5	18
Al Pacino	1	1	3	8	23
Teri Hatcher	1	1	2	4	11

At the ideal condition, i.e., considering the value of r as 1.13, we carried out the alias extraction process further. The window size in candidate alias extraction and w_c in context identification are set to 5 and 10 respectively, and only top 200 F-scoring lexical patterns are considered as suggested in [11]. After performing the filtering and cleaning tasks on extracted adjacent n-grams following the mentioned procedure, the set of candidate aliases are assigned three scores and aggregated as described in section 3.5. Figure 9 presents top 10 candidate aliases along with the score values calculated by the proposed system for a test case of *David Hasselhoff*. The three highlighted candidate aliases in the figure, *hoff*, *michael* and *michael knight* are the gold standards as defined in the dataset. First two of our extracted aliases match with the defined gold standards but the third ranking alias *david* does not exist in the gold standard set, whereas the fourth one is correctly extracted. Moreover, we see that the aggregated score for the candidate *hoff* is 0.903 which is much higher than the score 0.479 for *michael*, just after in the ranking order. This difference highlights how much popular is *hoff* than the remaining aliases.



Candidates	AS	SS	CS	Aliasness
hoff	1.620	0.709	0.670	0.903
michael	1.381	0.337	0.567	0.479
david	0.510	0.789	0.636	0.409
michael knight	1.583	0.083	0.596	0.375
knight	1.007	0.118	0.605	0.266
david michael	0.490	0.497	0.227	0.156
nick	0.316	0.153	0.581	0.107
hoff makes his first	0.561	0.272	0.175	0.100
knight rider	0.442	0.122	0.557	0.123
hoff makes	0.561	0.281	0.087	0.077

Figure 9: Test case scores for *David Hasselhoff*

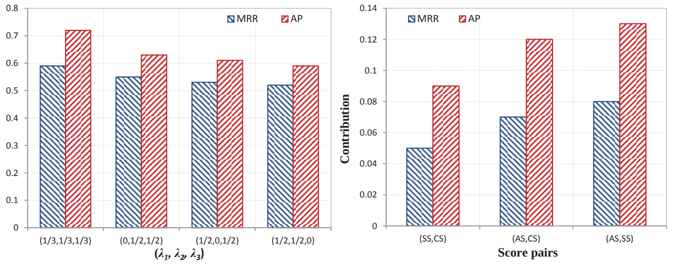
Table 8 presents the extracted aliases of four test cases by the proposed approach and compared with the gold standard aliases as well as those extracted by Bollegala *et al.* in [11]. We can see that our aliases are comparable to them as well.

The ranking of extracted alias names is based on the score aggregation formulated in equation 19, which uses three constants for controlling the affect of score pairs. We experimented by applying different values of these constants. Figure 10(a) presents the results obtained at some of these combinations in terms of MRR and AP. It thus shows the impact of varying the values of $(\lambda_1, \lambda_2, \lambda_3)$ on overall results. We found the best results at $\left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)$, and therefore finally settled with this. Figure 10(b) shows the contribution of each of the three score pairs, which is computed as the difference in results when the constant corresponding to a specific feature pair is set to zero. The score combination of assocScore and simScore contributes the most, whereas the combination of simScore and coocScore contributes the least, in the overall ranking.

Table 9 presents the overall performance of the proposed system on this dataset in terms of MRR and AP. The dataset here is free from the presence of multiple namesakes, i.e., all the webpages for each name refer to that specific person only. Therefore the namesake disambiguation task has been experimented on two other datasets, and the results in comparison to previous works have been shown in subsection 4.3. However, in this experiment too, our methodology shows good results and

Table 8: Comparison of extracted aliases

Real Name	Gold Standard Aliases	Top-3 Aliases by Bollegala *et al.* [11]	Top-3 Aliases by Proposed System
David Hasselhoff	hoff, michael knight, michael	hoff, michael knight, michael	hoff, michael, david
Courteney Cox	cece, lucy, dirt lucy, monica geller, monica	dirt lucy, lucy, monica	lucy, dirt lucy, monica
Al Pacino	sonny, alfredo james pacino, michael chorleone	michael chorleone, alfredo james pacino, alphonse pacino	michael chorleone, alfredo james pacino, michael
Teri Hatcher	hatch, susan mayer, susan, lois lane, lois	susan mayer, susan, mayer	susan mayer, susan, lois lane



(a) Results on varying Λ

(b) Score-pair contributions

Figure 10: Impact of $(\lambda_1, \lambda_2, \lambda_3)$ on overall results

when compared to the state-of-the-art method of Bollegala *et al.*, which do not consider namesake as an issue, our method performs comparably well. Although our MRR score 0.5938 is slightly lower than their score 0.6150, our AP score 0.7215 is considerably higher.

Table 9: Overall performance comparison

Approach	MRR	AP
Bollegala *et al.* [11]	**0.6150**	0.6865
Proposed Approach	0.5938	**0.7215**

5. A Move Towards Suspect Tracking on the Web

5.1. The Pervasive Social Media and its Psychological Impact Towards Cyber-crimes

The enhanced multi-media support of Web 2.0 and the ease of its access have led several new trends to brandish among the newer generations of people. One of the most dominating trends is the creation and maintenance of personal information on World Wide Web (WWW), and the use of this information as an identity to interact with others either publicly through the social media to discuss on topics of common interest or privately through e-mailboxes. The statistics of the most viewed websites shown in section 1 show the rapid attention of Web users towards social media to discuss on their topics of interest and exchange their like-minded thoughts. As a result everyday millions of user-generated messages are getting accumulated from the user comments.

Social interactions have always played crucial role in influencing personal lives by its psychological impact [24]. Uptil the last decade, these interactions remained confined only to real terrestrial societies, but these days the pervasive nature of social media is carrying them along with several new kind of relationships on to the Web to establish virtual online societies. Several studies are being performed on the psychological impact posed by these virtual community social ties on personal lives, in which it has been found that they can easily persuade the emotions of people towards any object, or issue or person [35, 1]. This impact depends very much on the psychological or mental status of people with whom the person is connected through the ties. Accordingly it can come up as a positive influence of emotions if those people are highly virtuous, or even negative influence of emotions, if they are morally and ethically down.

Unfortunately in this age of international terrorism and global drug smuggling, the latest trends in social media have also proliferated its use by various tech-savvy anti-social people for better communication among them and spreading their propaganda around the globe [46, 37, 4]. Due to their consistent efforts of persuasion for unethical negative causes, the probability of dominance of *negative affects* among the social media users are getting higher. This persuasion may even turn out to be violent to motivate them collaboratively to perform various kind of crimes, either digitally in the cyber-space or terrestrially in the real world. With the widespread growth of these organized and established fashion to commit crimes, the current investigation departments are lacking technologies to relate a person's *cyber-activities* to *terrestrial-activities* [22]. Although a number of cyber-crime threats are regularly being exposed, little is being done to protect against. The costliest form of such crime could be a transnational attack on computers and the information infrastructure to get a control over them. The proliferation of social media is thus acting as a catalyzing agent for persuasion towards cyber-crimes, that range from cyber-terrorism and drug-smuggling to cyber-bullying, pornography, and such other heinous acts [4].

5.2. Role of Alias Mining in Suspect Tracking through Social Media

Be it a cyber-crime or a terrestrial-crime, once a crime-suspect is found, the security intelligence organizations need to consider all possible platforms for investigation, e.g., travel details, telephone call records, activities on the Internet, and so on. Social media activities is a new add-on to this list, which may sometimes reveal a significant amount of crucial information as evidence of the suspect's involvement in a crime. In [19], Goodman highlighted the serious threat of cyber-crimes that we are heavily prone to, and has called an urgent requirement for counter cyber-crime measures. The prevalence of terrorism persuasion activities on the Web have well been noticed by the researchers and it is gaining subsequent attention to track them terrestrially as well as digitally. Wilson [46] exposed the pervasiveness of cyber-crimes and the botnets being used for its practice. He also analyzed the role of terrorism behind these crimes, and found it as a major reason of rising crimes. In [50], Zeng *et al.* analyzed the affect of social media on a naive web-user, and focusing on terrorism they also brought forth the violence in dark side of the Web (cyber-terrorism). While performing these unethical and immoral activities, very often the crime perpetrators use some alias names to represent themselves to their network partners. Sometimes it is done intentionally to hide their original identity, but it is also very common to use alias names just for the sake of simplicity. However, in crime cases, most of the time aliases are intentionally used as they enable to keep their identity as hidden. Identifying alias names of a crime-suspect on the Web is root problem for suspect tracking. Voss and Joslyn along with their team went through an exhaustive analysis of terrorist organizations and affiliated members related to 9/11 terrorist attack, FBI's most wanted list, and five important people of Al-Qaeda network [45]. Their data set was collected from open-sources of newspaper reports and various websites. A problem that acted as a major blockade in their analysis is the use of alias names by the terrorists in different sources, which had made it difficult to establish links between different incidents. Realizing the problem of alias mining for suspect analysis and investigation, Hsiung *et al.* [21] worked for detection of alias names from link data sets by employing a combination of orthographic and semantic information using various measures. The methodology proposed by us considers all kinds (unstructured / semi-structured) of textual data available on the Web, and it is based on a clue provided by the text-patterns that generally associate a real name with an alias name. Therefore, our system can identify an alias name only when it has been introduced along with the real name at least once. If somebody is using an alias name with an intention to hide the original identity, although the person himself or herself will never reveal the real name, but it may possibly be used by his or her network partners. Using this information our method would be able to find a matching text-pattern and thus identify the alias names. There are still many challenges yet to be tackled to identify alias names for a person on the Web.

6. Discussion

A major shortcoming that we observed in state-of-the-art alias mining approaches is that the possibility of the presence of multiple namesake individuals on the Web has remained a completely ignored issue [20, 10, 11, 6]. In spite of this, in [11] Bollegala *et al.* found impressive results in their experiments. Analyzing their dataset of English person names we found that all of the 50 persons in the dataset are renowned for their influential charisma. It makes them so popular on the Web to dominate over it and search engines rarely retrieve any page referring to any other namesake individual with the common name. Due to this predominance of these names on the Web, their approach found impressive results. But for a common name, say *John*, for which a number of persons with different identities exist on the Web, it would assume all of them as the same person and mix them making it very difficult to identify the actual aliases of the specific person. Another concern is that their system is heavily dependant on hyperlinks to generate the co-occurrence graph, which rarely appear in user-generated contents of social media. However, their approach is highly applicable for dominating personalities who have their own homepages, and specially when there also exist other pages referring to them. They used a set of 22 statistical features to train a ranking SVM, i.e., their system needs to be trained on realtime dataset, which is a cumbersome activity.

In contrast, we do not blindly accept all the webpages returned by the search engine as referring to a particular individual, rather we employ a content-based graphical disambiguation algorithm to separate them into different clusters of webpages corresponding to each namesake. It makes our system applicable to any name irrespective of its popularity on the Web. After getting the clusters, alias names are mined from each of them. The proposed alias mining algorithm is completely based on Web textual content, ignoring the occurrence of hyperlinks, that makes it applicable even for the user-generated contents of social media to incorporate real-time user views. In the feature set in [11], we observed that although they have a long list of features, many of these features have overlapping roles. For example, they have three page-count-based association measures as *WebDice*, *WebPMI* and *conditional probability*, but all of them highlight a common aspect which is the page-level co-occurrence popularity of the alias with the corresponding real name. Comparatively, our feature set is smaller in size but effective very much to capture the leading factors.

Our experimental results show the robustness of our approach. For webpage disambiguation, Dornescu *et al.* [16] have experimented their system using MCL, but their dataset is different than ours. We evaluated our system on the dataset that was also experimented in [23] and our disambiguation results are comparable with a computationally efficient approach. In graph generation, for each hyperlink they considered nodes for seed URLs in every level of domain which incurs heavy overhead by raising the number of nodes, as webpages generally contain a large number of hyperlinks in them. Also our network structure of the graph is much realistic. The computational complexity of our clustering algorithm MCL is $O(in^3)$, where i is the number of iterations until convergence. However during the process the matrix becomes sparse very quickly which urges for sparse matrix multiplication in $O(n^2)$ for expansion in later iterations. Finally, after the convergence the weakly connected components for cluster identification can be found in $O(n + m)$, where m is the number of links. A substantial part of the credit goes to the clustering technique that resolves the problem to specify the number of clusters to be generated beforehand. Instead of having a pre-determined number of clusters, MCL needs to have the pre-determined inflation parameter. Although prior experiment on representative sample datasets needs to be conducted for both of them for predetermination, there is an advantage with determination of inflation parameter over that of number of clusters directly. For a dataset, say $\{2, 3, 20, 21\}$, the optimal number of clusters could be 2 or 4 or even 1, depending on the required level of disambiguation, but not 3. The advantage with inflation parameter is that it decides just the level of disambiguation to select between 1, 2, and 4 clusters, as higher values of the parameter lead to stricter disambiguation producing clusters smaller in size but more in number. The level of required disambiguation isn't given the due concern when determining the number of clusters directly. Moreover, it is expected that for no value of the parameter, MCL would produce 3 clusters for this data, thereby automatically discarding the unsuitable number of clusters. The complexity and effectiveness of the task of determining the value of this parameter again depends on the size and quality of the representative sample. In this paper we do not focus on the technique to be employed for parameter determination, as it is more an application-specific problem. Nevertheless, determining the proper value is crucial for the following task.

The steps of candidate alias identification, feasibility analysis and their ranking employ very light-weight computations. The overall system has been evaluated on different datasets and our results are found comparatively well enough.

7. Conclusion and Future Work

In this paper, we presented a generic context-based approach for mining alias names of namesakes sharing a common name on the Web. Selection of the individual, for whom alias names are sought among the different namesakes, is left on part of the user to decide the one of interest. It addresses user-centric information extraction tasks on the Web, which currently face many challenges posed by the name ambiguities. The namesake disambiguation technique using Markov clustering makes the user free from the usual restriction of clustering techniques, and doesn't require to pre-determine the number of clusters to be generated. Rather it is determined dynamically using the experimentally obtained inflation parameter, which, unlike determining number of clusters directly, is based on the required level of disambiguation making the task much easier. The proposed statistical measures for alias mining capture the prominent information from three diverse but salient aspects, *associativity*, *similarity* and *co-occurrence*, in an efficient and effective manner. The overall approach has been evaluated on different standard datasets to establish its efficacy. We also introduced the significance of alias mining to deal with non-social users on the Web to track their malicious activities on various social media platforms. Due to a rapid growth of radical and malicious activities on the Web these days, it opens up an area of concern for applying user-centric text mining techniques to monitor these unethical activities and track the suspects.

Some of the promising directions for future research are explorations of implicit relationships among web users, creation and employment of external knowledge bases to enable better disambiguation, and to explore and discover some way to extract candidate aliases without depending on the available text pattern dataset. Since the performance of existing NERs is not much satisfactory in some cases for the webpage texts, working for an alternate and better mechanism for document graph generation could be a good task for future work. In future, we intend to work on these problems and apply text mining techniques to aid in suspect tracking on the Web.

Acknowledgment

The authors would like to thank Prof. Mohammed J. Zaki (RPI, New York) for our productive discussions with him and his extensive support throughout this work, and Dr. Danushka Bollegala (University of Tokyo, Japan) for sharing the text-patterns discovered by them in their research on alias identification. The authors would also like to thank King Abdulaziz City for Science and Technology (KACST) and King Saud University for their support. This work has been financially supported by KACST under the NPST project number `11-INF1594-02`.

[1] A. Abbasi, H. Chen, S. Thoms, T. Fu, Affect analysis of web forums and blogs using correlation ensembles, IEEE Trans. on Knowl. and Data Eng. 20 (9) (2008) 1168–1180.

[2] M. Abulaish, T. Anwar, A supervised learning approach for automatic keyphrase extraction, International Journal of Innovative Computing, Information and Control 8 (11) (2012) 7579–7601.

[3] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions, IEEE Trans. on Knowl. and Data Eng. 17 (6) (2005) 734–749.

[4] T. Anwar, M. Abulaish, Identifying cliques in dark web forums- an agglomerative clustering approach, in: Proc. of the 2012 IEEE Int'l Conf. on Intelligence and Security Informatics, ISI'12, IEEE Computer Society, 2012, pp. 171–173.

[5] T. Anwar, M. Abulaish, An mcl-based text mining approach for namesake disambiguation on the web, in: Proc. of the 2012 IEEE/WIC/ACM int'l conf. on Web Intelligence 2012, WI'12, IEEE Computer society, 2012, pp. 40–44.

[6] T. Anwar, M. Abulaish, K. Alghathbar, Web content mining for alias identification: A first step towards suspect tracking, in: Proc. of the 9th IEEE Int'l Conf. on Intelligence and Security Informatics, ISI'11, IEEE Computer Society, 2011, pp. 195–197.

[7] J. Artiles, J. Gonzalo, S. Sekine, Weps 2 evaluation campaign: Overview of the web people search clustering task, in: Proc. of the 2nd Web People Search Evaluation Workshop on 18th WWW Conference, WePS'09, 2009.

[8] A. Bagga, B. Baldwin, Entity-based cross document co-referencing using vector space model, in: Proc. of the 17th Int'l Conf. on Computational Linguistics, COLING'98, ACM, 1998, pp. 79–85.

[9] R. Bekkerman, A. McCallum, Disambiguating web appearances of people in a social network, in: Proc. of the 14th Int'l Conf. on World Wide Web, WWW'05, ACM, 2005, pp. 463–470.

[10] D. Bollegala, T. Honma, Y. Matsuo, M. Ishizuka, Mining for personal name aliases on the web, in: Proc. of the 17th Int'l Conf. on World Wide Web, WWW'08, ACM, 2008, pp. 1107–1108.

[11] D. Bollegala, Y. Matsuo, M. Ishizuka, Automatic discovery of personal name aliases from the web, IEEE Trans. on Knowl. and Data Eng. 23 (6) (2011) 831–844.

[12] P. A. Chirita, C. S. Firan, W. Nejdl, Personalized query expansion for the web, in: Proc. of the 30th Ann. Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, SIGIR'07, ACM, New York, NY, USA, 2007, pp. 7–14.

[13] P. Cimiano, S. Handschuh, S. Staab, Towards the self-annotating web, in: Proc. of the 13th Int'l Conf. on World Wide Web, WWW'04, ACM, New York, NY, USA, 2004, pp. 462–471.

[14] S. Cucerzan, Large-scale named entity disambiguation based on wikipedia data, in: Proc. of the 2007 Jt. Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), Association for Computational Linguistics, 2007, pp. 708–716.

[15] M. Deshpande, G. Karypis, Item based top-n recommendation algorithms, ACM Trans. on Information Systems 22 (1) (2004) 143–177.

[16] I. Dornescu, C. Orasan, T. Lesnikova, Cross-document coreference for weps, in: CLEF LABs: Web People Search (WePS3), 2010.

[17] A. A. Ferreira, M. A. Gonçalves, J. M. Almeida, A. H. F. Laender, A. Veloso, A tool for generating synthetic authorship records for evaluating author name disambiguation methods, Information Sciences 206 (2012) 42–62.

[18] N. Godbole, M. Srinivasaiah, S. Skiena, Large-scale sentiment analysis for news and blogs, in: Proc. of the Int'l Conf. on Weblogs and Social Media, ICWSM'07, 2007.

[19] M. Goodman, International dimensions of cybercrime, in: Cybercrimes: A Multidisciplinary Analysis, Springer-Verlag, 2010.

[20] T. Hokama, H. Kitgawa, Extracting mnemonic names of people from the web, in: Proc. of the 9th Int'l Conf. on Asian Digital Libraries, ICADL'06, Springer-Verlag, 2006, pp. 121–130.

[21] P. Hsiung, A. Moore, D. Neill, J. Schneider, Alias detection in link data sets, in: Proc. of the Int'l Conf. on Intelligence Analysis, 2004.

[22] F. Iqbal, H. Binsalleeh, B. C. M. Fung, M. Debbabi, A unified data mining solution for authorship analysis in anonymous textual communications, Information Sciences 231 (2013) 98–112.

[23] D. V. Kalashnikov, Z. Chen, S. Mehrotra, R. Nuray-Turan, Web people search via connection analysis, IEEE Trans. on Know. and Data Eng. 20 (11) (2008) 1550–1565.

[24] I. Kawachi, L. F. Berkman, Social ties and mental health, Journal of Urban Health 78 (3) (2001) 458–467.

[25] R. Kosala, H. Blockeel, Web mining research: A survey, SIGKDD Explor. Newsl. 2 (1) (2000) 1–15.

[26] E. Lefever, T. Fayruzov, V. Hoste, M. De Cock, Clustering web people search results using fuzzy ants, Information Sciences 180 (17) (2010) 3192–3209.

[27] A. Lenhart, K. Purcell, A. Smith, K. Zickuhr, Social media & mobile internet use among teens and young adults, Tech. rep., PewResearch Center, http://pewinternet.org/Reports/2010/Social-Media-and-Young-Adults.aspx (2010).

[28] Y. Li, N. Zhong, Web mining model and its applications for information gathering, Knowledge-Based Systems 17 (5-6) (2004) 207–217.

[29] J. Lin, The web as a resource for question answering: Perspectives and challenges, in: Proc. of the 3rd Int'l Conf. on Natural Language Resources and Evaluation, LREC'02, 2002.

[30] C. Long, L. Shi, Web person name disambiguation by relevance weighting of extended feature sets, in: Proc. of the CLEF (Notebook Papers/LABs/Workshops), 2010.

[31] V. Lopez, M. Pasin, E. Motta, Aqualog: An ontology-portable question answering system for the semantic web, The Semantic Web: Research and Applications 3532 (2005) 135–166.

[32] Y. Matsuo, J. Mori, M. Hamasaki, K. Ishida, T. Nishimura, H. Takeda, K. Hasida, M. Ishizuka, Polyphonet: An advanced social network extraction system from the web, in: Proc. of the 15th Int'l Conf. on World Wide Web, WWW '06, ACM, New York, NY, USA, 2006, pp. 397–406.

[33] K. J. Mitchell, J. Wolak, D. Finkelhor, Trends in youth reports of sexual solicitations, harassment and unwanted exposure to pornography on the internet, Journal of Adolescent Health 40 (2) (2007) 116–126.

[34] B. Pang, L. Lee, Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (2008) 1–135.

[35] R. E. Petty, D. T. Wegener, Attitude change: Multiple roles for persuasion variables, Social Psychology 1 (1) (1998) 1–78.

[36] J. Piskorski, K. Wieloch, M. Sydow, On knowledge-poor methods for person name matching and lemmatization for highly inflectional languages.

[37] J. Qin, Y. Zhou, H. Chen, A multi-region empirical study on the internet presence of global extremist organizations, Information Systems Frontiers 13 (1) (2011) 75–88.

[38] G. D. M. Rennie, T. Jaakola, Using term informativeness for named entity detection, in: Proc. of the 28th Ann. Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, 2005.

[39] G. Salton, Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer, Addison-Wesley Longman Publishing Co., Boston, MA, USA, 1989.

[40] M. Sanderson, Word sense disambiguation and information retrieval, in: Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR'94, 1994, pp. 142–151.

[41] D. Sculley, G. M. Wachman, Relaxed online svms for spam filtering, in: Proc. of the 30th Ann. Int'l ACM SIGIR Conf. on Research and Development in Information Retrieval, ACM, 2007, pp. 415–422.

[42] J. Srivastava, R. Cooley, M. Deshpande, P. N. Tan, Web usage mining: Discovery and applications of usage patterns from web data, SIGKDD Explorations Newsletter 1 (2) (2000) 12–23.

[43] K. sugiyama, K. Hatano, M. Yoshikawa, Adaptive web search based on user profile constructed without any effort from users, in: Proc. of the 13th Int'l Conf. on World Wide Web, WWW'04, ACM, NY, USA, 2004, pp. 675–684.

[44] S. Van Dongen, A cluster algorithm for graphs, Ph.D. thesis, University of Utrecht (2000).

[45] S. Voss, C. Joslyn, Advanced knowledge integration in assessing terrorist threats, Tech. rep., LAUR 02-7867, http://www.au.af.mil/au/awc/awcgate/lanl/knowledge_integration.pdf (2002).

[46] C. Wilson, Botnets, cybercrime, and cyberterrorism: Vulnerabilities and policy issues for congress, Tech. rep., Congressional Research Service, http://www.fas.org/sgp/crs/terror/RL32114.pdf (2008).

[47] S. T. Wu, Y. Li, Y. Xu, B. Pham, P. Chen, Automatic pattern-taxonomy extraction for web mining, in: Proc. of the 2004 IEEE/WIC/ACM Int'l Conf. on Web Intelligence, WI'04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 242–248.

[48] X. Wu, L. Zhang, Y. Yu, Exploring social annotations for the semantic web, in: Proc. of the 15th Int'l Conf. on World Wide Web, WWW'06, ACM, New York, NY, USA, 2006, pp. 417–426.

[49] M. J. Zaki, W. Meira Jr., Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, 2014.

[50] S. Zeng, M. Lin, H. Chen, Dynamic user-level affect analysis in social media: Modeling violence in the dark web, in: Proc. of the IEEE Int'l Conf. on Intelligence and Security Informatics, ISI '11, IEEE Computer Society, 2011, pp. 1–6.

[51] L. Zhang, J. Zhu, T. Yao, An evaluation of statistical spam filtering techniques, ACM Trans. on Asian Language Information Processing (TALIP) 3 (4) (2004) 243–269.

[52] Z. Zhang, H. Lin, K. Liu, D. Wu, G. Zhang, J. Lu, A hybrid fuzzy-based personalized recommender system for telecom products/services, Information Sciences 235 (2013) 117–129.