

An MCL-Based Text Mining Approach for Namesake Disambiguation on the Web

Tarique Anwar

Center of Excellence in Information Assurance
King Saud University, Riyadh, Saudi Arabia
Email: tAnwar.c@ksu.edu.sa

Muhammad Abulaish, *SMIEEE*

Center of Excellence in Information Assurance
King Saud University, Riyadh, Saudi Arabia
Email: mAbulaish@ksu.edu.sa

Abstract—In this paper, we propose a Markov CLustering (MCL) based text mining approach for namesake disambiguation on the Web. The novelty of the proposed technique lies in modeling the collection of webpages using a weighted graph structure and applying MCL to crystalize it into different clusters, each one containing the webpages related to a particular namesake individual. The proposed method focuses on three broad and realistic aspects to cluster webpages retrieved through search engines – *content overlapping, structure overlapping, and local context overlapping*. The efficacy of the proposed method is demonstrated through experimental evaluations on standard datasets.

Index Terms—Text mining, Web content mining, Web people search, Namesake disambiguation, Markov clustering.

I. INTRODUCTION

Due to easy and cheap accessibility of the World Wide Web (WWW), it has become a fascinating trend for Web users to create and maintain personal profiles and use them to interact with others in the cyberspace. The number of Web users is found to increase rapidly and a parallel growth is seen in the pervasiveness of social media into our personal lives through weblogs, forums, and social networking and video sharing sites. According to the statistics of *DoubleClick AdPlanner*¹, social media sites are among the top ranked websites with the largest number of visitors and page views. The user-generated content using social media has opened up a gateway for several entity-centric research tasks that focus on user activities. Research areas like sentiment analysis and predicting user behavior and trends have received a major thrust with the advent of social media. It is providing a rich set of features on a platform for the business community to analyze their business strategies with respect to the concerns of end-users expressed by them in the form of user-generated content. Moreover, now-a-days even common Web users are getting more curious to know about others, with whom they are somehow related, and consequently *people search* on the Web using name is found as the most common activity of Web users. As reported in [1], about 30 percent of search-engine queries include person names. Thus, entity-centric search seems to play a key role in mapping information to persons' identity.

On analysis, we found that there are two major problems associated with entity-centric search; particularly, if the entity is considered as a person name. First, it is very common to have multiple names for an entity. For example, *Mumbai* (a city in India) is very often called by its anglicised name *Bombay* and consequently a number of webpages contains this word. In contrast, the other problem is the representation of different entities using a same name. In this case, there exists an entity on a page that could have two or more different interpretations. For example, the entity “*Puma*” can refer to multiple meanings. It can be a Brazilian brand of sports car, *Cagaur* (a large cat), *AMD Puma* (a mobile computing platform), *Puma* (a local language of Nepal), or a person *surname*. This ambiguity needs to be resolved for refining entities for a better search result. Both of these entity-centric search problems are also faced for *person name search*. Web people search is completely a person name centric problem, and while querying the Web to search for someone, a traditional search engine retrieves a blend of all the webpages containing the queried name, irrespective of the specific individual for whom the user is looking. And, it becomes users' responsibility to discard irrelevant webpages referring to some other namesake sharing the common name, which is quite a tedious job to go through each and every page manually. Table I shows text snippets from five webpages retrieved for the name *Michael Clark* by Google search engine, where each snippet refers to a specific namesake individual.

Although, the existing search engines facilitate searching people on the Web, they are not competent enough to care for namesake disambiguation before presenting the results to users. However, this limitation can be tackled up to some extent by applying text mining techniques to process the search results obtained from the search engines. In this paper, we propose a namesake disambiguation process that receives the list of webpages returned by a search engine in response to a given person name search query and cluster them in different groups, each one representing a particular namesake individual. Starting with the transformation of retrieved webpages into a weighted graph, the namesake disambiguation approach applies Markov Clustering (MCL) algorithm to crystalize it into different subgraphs, each one containing the pages related to a particular namesake. The novelty of the proposed approach lies in the realistic graph generation as well as the

¹<http://www.google.com/adplanner/>

TABLE I
RETRIEVED WEBPAGES BY GOOGLE FOR A PEOPLE SEARCH QUERY
“MICHAEL CLARK”

Source	Snippet
http://en.wikipedia.org/wiki/Michael_Clark_(boxer)	...Michael Clark (b. July 10, 1973, Columbus, Ohio) is a professional boxer . He has twice won the IBC lightweight title...
http://en.wikipedia.org/wiki/Michael_Clark_(Canadian_politician)	...Michael Clark (born: 12 May 1861 Belford, Northumberland, England - died: July 29, 1926 Olds, Alberta) was a Canadian physician and politician from Alberta, Canada...
http://en.wikipedia.org/wiki/Michael_Clark_(dancer)	...Michael Clark was born in Aberdeen and began traditional Scottish dancing at the age of four. In 1975 he left home to...
http://en.wikipedia.org/wiki/Michael_Clark_(British_politician)	...Michael Clark (born 8 August 1935) is a British Conservative politician ...
http://www.humanism.org.uk/about/philosophers/members	...Michael Clark taught Philosophy at the University of Nottingham . He read Philosophy and Psychology at Oxford and moved to Nottingham after five years...

clustering technique that does not need to input the number of clusters, which is usually a difficult task if the number of classes in the given dataset is not known.

II. RELATED WORK

Due to its profound applicability, research on Web people search and specifically, on person name disambiguation has gained adequate attention from the scientific community. In [2], to disambiguate Web appearances, the authors followed the hypotheses that webpages referring to the same person have higher chances to be inter-linked and therefore exploited the link structure of the webpages. They also integrated this to the content-based analysis and applied Agglomerative/Conglomerative Double Clustering (A/DC) for classification. The work in [3] used a rich set of features comprising tokens, named entities, host names and URLs for clustering webpages using hierarchical agglomerative clustering method. Kalashnikov *et al.* came up with their approach in [4], in which an undirected weighted graph is generated with nodes representing webpages and the entities (named entities, URLs and email ids) extracted from them. A similarity value is calculated between every pair of nodes representing webpages in the graph by combining *cosine similarity* based on *TF-IDF* measure and *connection strength* between each pair of nodes, which is calculated as an aggregation of the weights of *L*-shortest paths between the nodes. They have applied correlation clustering to crystalize the graph. In [5], Yoshida *et al.* designed a two-stage clustering algorithm for name disambiguation and in a recent work [6] they proposed a bootstrapping approach.

Since 2007 a workshop named as Web People Search (WePS) is being conducted to provide a common platform to discuss the issues associated with entity-centric information search on the Web. WePS-3, the most recent one, was held in 2010 including 34 participating teams with an interest in the clustering task, out of which only 8 teams submitted results till the last run. Smirnova and Trousse [7] applied a two-stage clustering algorithm based on some content-based features and also analyzed a set of patterns in the Web graph. In [8], the authors designed four document features and six pairwise similarity measures and used them to train a pairwise model. The model is designated to predict whether two documents

refer to same person or not. For the clustering task they used HAC and MCL. There are few Web search engines (e.g., Clusty, Vivisimo, Zoominfo, etc.) that are either publicly available or on commercial basis to provide the functionality for people search on the Web. Although, they retrieve results that are grouped into different clusters, their objective behind clustering is not to disambiguate namesakes, rather to group webpages together that are related to the same topic.

III. PROPOSED METHOD

In this section, we present the design of our proposed MCL-based text mining approach for namesake disambiguation as shown in figure 1. After analyzing the existing literature on web people search, we found that webpage disambiguation for the namesakes is an important task to perceive reliable and helpful information during Web people search. We have considered three prime factors that are solely responsible to lead a human being to disambiguate namesake webpages manually: i) *content overlap* to determine the extent of match between the content of different webpages, ii) *structure overlap* to determine the extent of match between the hyperlinks of different webpages, and iii) *local context overlap* to determine the extent of match between local context of the entity names appearing in different webpages. The proposed method being centric to these three factors to resolve namesake ambiguities on the Web, goes through five major steps – *webpage retrieval and content extraction* to retrieve webpages for the given person name and to extract relevant text blocks from them; *entity extraction* to extract entities from retrieved webpages; *graph generation* to transform the webpages into a semi-structured format using the identified entities; *clustering* to group the related webpages together; and finally *profile summary generation* to provide a summarized view of the identified information components for each individual namesake. Further details about each of these functionalities are presented in the following subsequent sub-sections.

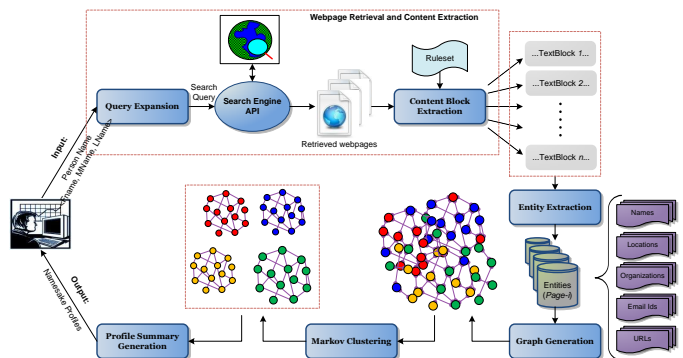


Fig. 1. Workflow of the proposed namesake disambiguation method

A. Webpage Retrieval and Content Extraction

In this step, webpages related to a given person name are crawled from the Web using Google API and stored on a local machine for further processing. Generally, people are habitual

to use their names in different forms, and to get rid of this problem rather than limiting our search to exact matches for the given name we relax the search criteria to include even the partial matches and filtered out the irrelevant ones locally following some heuristics. Due to semi-structured nature of the Web, very often the retrieved webpages are found too noisy being full of advertisements and containing many incomplete sentences, which generally increase the false positive and false negative rates of named entity recognizer. Therefore, we have designed a set of tag-based heuristic rules and used them to restrict incomplete sentences and those used for advertisement purpose from further considerations.

B. Entity Extraction

Generally, in a webpage not all provided information are important to locate the person to whom the page refers. Using a set of keyphrases from within the pages is a better choice to reduce the problem space, but reducing it further to only the named entities makes it most effective. Therefore, in line with the work proposed in [1], we consider five basic entity types (*person*, *place*, *organization*, *email ids*, and *hyperlinks*) to conceptualize webpages and use them as a basis to differentiate among different namesakes sharing a common name. First four of these entities capture *content-overlapping*, whereas the last one captures the *structural overlapping* among different webpages.

C. Graph Generation

In this step, the webpages are transformed into an undirected-weighted graph by adding all webpages and the unique entities extracted from them as nodes. Regardless to the number of webpages containing an entity e , a single node for e is added in the graph to represent all its appearances in those webpages. For each URL entity, one extra node is added that refers to the *seed URL* of the actual hyperlink. Seed URLs are also added for each of the email ids belonging to the respective domain. Links between different pairs of nodes are created using three different types of relationships – *entity-entity* relationship, *entity-page* relationship, and *page-page* relationship. For each type of relationship, a weight calculation mechanism is proposed to assign association weights to links. The edge creation and weight-calculation process is further detailed below:

- For each pair of entities e_i and e_j that has co-occurrence at the sentence-level, a link between them is created and its weight $\omega(e_i, e_j)$ is calculated using equation 1, which is defined as the normalization of the total no. of co-occurrences of e_i and e_j at sentence-level in the whole set of documents D .

$$\omega(e_i, e_j) = \frac{\sum_D \text{freq}(e_i, e_j)}{\max_{p,q,p \neq q} \{\sum_D \text{freq}(e_p, e_q)\}} \quad (1)$$

- For each entity e_i extracted from a document d_j , a link between e_i and d_j is created and its weight is calculated using equation 2, which is defined as the ratio of the

frequency count of e_i in d_j to the maximum frequency count of entities belonging to the class of e_i in d_j .

$$\omega(e_i, d_j) = \frac{\text{freq}(e_i, d_j)}{\max_{\text{type}(e_i)} \{\text{freq}(e_k, d_j)\}} \quad (2)$$

- For seed URL s_i , representing full URLs and/or e-mail ids, present in a document d_j , a link between s_i and d_j is created and its weight is calculated using equation 3, which is defined as the ratio of the frequency count of s_i in d_j to the maximum frequency count of entities in this category in d_j , multiplied by a *biasness control* factor where h is the no. of unique hyperlinked URLs associated with s_i .

$$\omega(s_i, d_j) = \frac{\text{freq}(s_i, d_j)}{\max_p \{\text{freq}(s_p, d_j)\}} \times \frac{1}{\log_2 h + 1} \quad (3)$$

- For each full URL u_i , a link is created between u_i and the seed URL s_j derived from u_i , and its weight is assigned as 1, i.e., $\omega(u_i, s_j) = 1$.
- Email addresses play a key role to identify an individual and generally people have their email ids in the domain of their organization’s website. To establish this relation, for each email id l_i , a link is created between l_i and the seed URL s_j derived from l_i , and its weight is assigned as 1, i.e., $\omega(l_i, s_j) = 1$.
- We also model the relationship between each pair of documents, which is defined as the degree of similarity between them. This similarity is measured in two different directions – *content similarity* and *local context similarity*. For content similarity, we consider the set of m retrieved documents as a corpus, $D = \{d_1, d_2, \dots, d_m\}$, and the set of unique entities extracted from D as the set of n distinct terms, $T = \{t_1, t_2, \dots, t_n\}$ to construct a *term-document matrix* $\Omega_{(n \times m)}$ with its values as the *TF-IDF* measures. Then *content similarity* C is measured as the cosine similarity between each pair of webpage nodes using equation 4.

$$C(d_i, d_j) = \frac{\sum_{l=1}^n \Omega_{(l,i)} \times \Omega_{(l,j)}}{\sqrt{\sum_{l=1}^n \Omega_{(l,i)}^2} \sqrt{\sum_{l=1}^n \Omega_{(l,j)}^2}} \quad (4)$$

On the other hand, *local context similarity* considers highly informative and fundamental English terms locally that present a background knowledge about the person being described. Therefore, for each webpage d_i , we generate a set of words X_i within a window size of w for each appearance of the person names using a simple pattern matching approach. Based on the *TF-IDF* measures of the identified context words, a *context-document matrix* $\Gamma_{(n \times m)}$ is constructed and *local context similarity* $L_{(d_i, d_j)}$ is defined as the cosine similarity between each pair of documents d_i and d_j . Finally, the weight for the link between each pair of webpages is calculated using equation 5, where $0 \leq \mu \leq 1$.

TABLE II
OVERALL RESULT SUMMARY ON TWO DIFFERENT DATASETS

Approach	Dataset [2]		Dataset [11]	
	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
Improvement	4.0	4.4	4.6	3.7

$$\omega(d_i, d_j) = \mu \times C(d_i, d_j) + (1 - \mu) \times L(d_i, d_j) \quad (5)$$

D. Clustering and Profile Summary Generation

The state-of-the-art solutions for this problem, as discussed in section II, have found graph-based clustering techniques to be the most successful because of its nature [4] [9]. Once the graph is generated for the complete set of webpages, we apply *Markov Clustering* (MCL) [10] on it which transforms G into a directed graph G_i with several weakly connected components, each one resulting into a separate cluster. MCL [10] is a graph clustering technique based on simulating a random walk on a weighted graph. It considers transition from one node to another within a cluster as much more likely than those in different clusters, taking into account the weight of their links. It accepts the adjacency matrix $A_{(n \times n)}$ of a graph G as an input and starts working by adding loops or self-edges to A 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 clustering process proceeds in an iterative manner interleaving matrix expansion by multiplying with itself and inflation using equation 6 until the transition matrix M_i converges, i.e., until the difference between the transition matrix from two successive iterations falls below some threshold $\theta \geq 0$.

The generated matrix M_i , after its convergence in the i th iteration, results into a directed graph with weakly connected components in it. The nodes having values greater than zero in the diagonal, i.e. $m_{(p,p)} > 0$, are called as *attractors* of the corresponding clusters. All the other nodes having a link with the attractors are attracted towards them and are included in the respective clusters. Once the clusters are determined through MCL, each of them is considered as a particular namesake and its profile is generated using the information components extracted from the webpages belonging to that cluster.

$$\xi(M, r) = \left\{ \frac{(m_{(p,q)})^r}{\sum_{a=1}^n (m_{(p,a)})^r} \right\}_{p,q=1}^n \quad (6)$$

IV. EXPERIMENTAL RESULTS

Since 2007 three open workshop tasks on Web People Search (WePS) have been conducted by the NLP group at UNED and they were actively participated by several teams from different research institutions, showing a global attention towards it. We carried out experiments on two different datasets – the dataset used by Bekkerman and McCallum in [2] and the WePS-2 test dataset. The Bekkerman dataset consists of text files created from webpages of 12 different person names which are primarily the names of SRI employees and professors from different universities, and the WePS-2 test dataset comprises of webpages for 30 person names, with 10 out of them collected from Wikipedia, another 10 from ACL’06 and the rest from US Census. For evaluation purposes,

the participants of WePS-1 used the metric $F_{\alpha=0.5}$ (or F_P^2 at $\alpha = 0.5$), as a measure for the quality of their systems. The participants of WePS-2 introduced another metric $F_{B-cubed}$ (or F_B^3) measure for the evaluation of their systems.

The overall result summary of our experiments conducted over the above-mentioned datasets is presented in table II. The first row presents the results at an earlier stage which doesn’t consider *local context overlapping* (LO) and *seed URL*, i.e., the value of μ in equation 5 is set to 1 and no extra node is added for the seed URL in case of hyperlinks and email ids. The second row presents the final result when μ is set to 0.5 and seed URL is included as a node in the graph. The last row showing the improvement values highlights the relevance of newly added features. In the experiment on Bekkerman dataset, after extracting entities by Stanford NER and HTML parser APIs embedded in the self coded rules, an undirected-weighted graph is generated as mentioned in section III-C. Finally, MCL is applied on the adjacency matrix of the generated graph with varying values of the inflation parameter r , setting the threshold value $\theta = 0.001$ and μ in equation 5 to 0.5. Table III shows the obtained results by our method, where we can see the values of F_B and F_P at $\alpha = 0.5$ for 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 their content, which seems very much realistic.

Figure 2 presents a comparative view of the no. of clusters generated for four different person names on varying values of r . It can be observed that the number of clusters increases with increasing value of r , however the exact no. of clusters cannot be predicted beforehand. It can also be observed that the number of clusters generated for *Tom Mitchell* grows much more rapidly than *Adam Cheyer*. For the value of r as 1.1, it generates 5 and 1 clusters for them respectively, whereas at $r = 1.2$ these values rise to 52 and 7 making a huge difference. The reason behind this difference is due to varying nature of

²If C is the set of clusters generated by the automated system and L is the gold standard set, then $purity = \sum_i \frac{|C_i|}{n} \max Precision(C_i, L_j)$ and $inversepurity = \sum_i \frac{|L_i|}{n} \max Precision(L_i, C_j)$. F_P is calculated as their harmonic mean.

³For each element (or post), i , precision and recall values are computed individually as $precision_i = \frac{C_i \cap L_i}{C_i}$ and $recall_i = \frac{C_i \cap L_i}{L_i}$. The average B-cubed precision and recall are computed as the mean of individual values. F_B is calculated as their harmonic mean.

TABLE III
RESULTS ON BEKKERMAN DATASET

Person name	$r = 1.10$	$r = 1.13$	$r = 1.15$	$r = 1.17$	$r = 1.20$
	F_B/F_P	F_B/F_P	F_B/F_P	F_B/F_P	F_B/F_P
Adam Cheyer	99.0/99.0	98.1/98.6	98.1/98.6	91.6/91.2	87.3/87.9
William Cohen	77.6/84.0	84.1/90.2	88.2/92.8	72.3/80.6	56.8/72.0
Steve Hardt	78.1/81.6	82.4/85.7	75.6/77.1	66.0/71.3	49.8/62.3
David Israel	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	98.4/98.5	90.7/91.1	88.4/88.7
Bill Mark	57.6/73.0	75.8/86.2	70.6/78.3	52.8/60.4	31.5/42.5
Andrew McCallum	55.3/60.1	68.7/71.6	82.5/84.8	73.9/77.1	57.8/65.2
Tom Mitchell	41.0/45.5	48.4/51.7	51.8/62.5	82.7/86.3	74.8/78.1
David Mulford	73.5/80.7	76.9/85.1	84.5/88.4	71.6/77.9	52.5/63.7
Andrej Ng	44.6/49.3	51.7/63.2	73.6/77.8	78.1/81.4	60.3/64.9
Fernando Pereira	65.2/68.0	72.4/77.5	68.8/72.3	54.7/59.2	39.9/48.6
Lynn Voss	38.4/47.7	53.1/67.3	62.1/69.6	71.0/77.9	58.6/60.8
Average	65.7/71.4	73.0/78.9	77.2/81.4	73.5/78.1	60.1/67.1

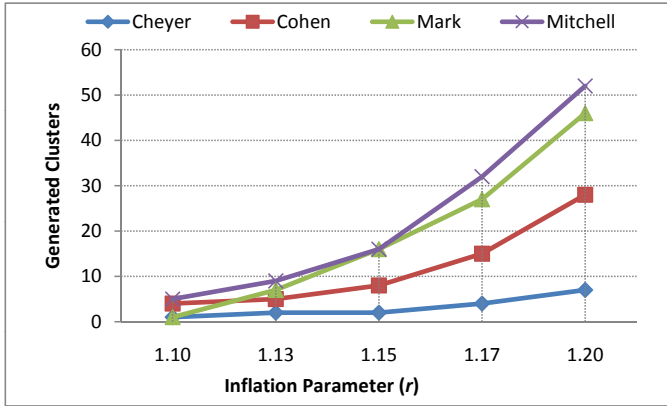


Fig. 2. No. of generated clusters for different values of r

content in their respective webpages. Moreover, we can see in table III that the best result for 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 1.1, whereas for *David Israel* the best result is found at 1.17. Focusing on the complete dataset, we see that the highest values for F_B/F_P are concentrated around $r = 1.15$, with best values for three names at 1.13, another three at 1.15 and another four at 1.17. Hence, in our approach calculating the average F-measures of all individual values, we found the best result ($F_B = 77.2\%$ and $F_P = 81.4\%$) when r is set to 1.15. On analysis, we found that the system shows better results for the value of r at which the no. of generated clusters are closer to the no. of manually annotated categories. Due to this reason, for $r > 1.15$ the performance starts degrading as the no. of generated clusters are much larger.

In [8], Dornescu *et al.* experimented their system using MCL, however their test dataset was different than ours. In [4], the authors evaluated their system on the same dataset as ours. Although their system is outperforming our approach, but we have come up with a computationally efficient and unsupervised approach using Markov clustering that doesn't need the number of clusters to be specified apriori. In graph generation, for each hyperlink they have considered nodes for seed URLs in every level of domain, which incurs heavy overhead by raising the number of nodes. Also our network structure of the

graph is much realistic. In their approach, the computational complexity of the connection strength calculation by adding up weights of L -shortest paths between two nodes in the graph is very high. The computational complexity of the clustering algorithm used in our approach is $O(in^3)$, where i is the number of iterations until convergence. However, during the clustering process the matrix becomes sparse very quickly and sparse matrix multiplication can be used in later iterations which has a complexity of $O(n^2)$. The convergence is usually achieved in very few iterations. Finally, after the convergence the weakly connected components for the clusters can be found in $O(n + m)$, where m is the number of links.

V. CONCLUSION

This paper presents an MCL-based document clustering approach for namesake resolution on the Web. The proposed approach first models the retrieved webpages into an undirected-weighted graph and applies MCL to crystalize it into different clusters, each one representing a particular namesake individual. On analysis we found that the computational complexity of the proposed method is quite satisfactory in comparison to other state-of-the-art techniques.

ACKNOWLEDGMENT

The authors would like to thank King Abdulaziz City for Science and Technology (KACST) and King Saud University for their support. This work has been funded by KACST under the NPST project number 11-INF1594-02.

REFERENCES

- [1] D. Bollegala, Y. Matsuo, and M. Ishizuka, "Automatic discovery of personal name aliases from the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 831–844, 2011.
- [2] R. Bekkerman and A. McCallum, "Disambiguating web appearances of people in a social network," in *Proceedings of the 14th International Conference on World Wide Web (WWW 2005)*, 2005.
- [3] E. Elmacioglu, Y. Fan, T. Su, Y. Min-yen, and K. D. Lee, "Psnus: Web people name disambiguation by simple clustering with rich features," in *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*. Association for Computational Linguistics, Jun. 2007, pp. 268–271.
- [4] D. V. Kalashnikov, Z. Chen, S. Mehrotra, and R. Nuray-Turan, "Web people search via connection analysis," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 11, pp. 1550–1565, Nov. 2008.
- [5] M. Yoshida, M. Ikeda, S. Ono, I. Sato, and H. Nakagawa, "Person name disambiguation on the web by two-stage clustering," in *Proceedings of the 18th World Wide Web Conference*, Apr. 2009.
- [6] —, "Person name disambiguation by bootstrapping," in *Proceedings of 33rd Annual ACM SIGIR Conference on Research and Development in Information Retrieval*, Jul. 2010.
- [7] E. Smirnova, K. Avrachenkov, and B. Trousse, "Using web graph structure for person name disambiguation," in *Proceedings of the CLEF (Notebook Papers/LABs/Workshops)*, 2010.
- [8] I. Dornescu, C. Orasan, and T. Lesnikova, "Cross-document coreference for weps," in *Proceedings of the CLEF (Notebook Papers/LABs/Workshops)*, 2010.
- [9] C. Long and L. Shi, "Web person name disambiguation by relevance weighting of extended feature sets," in *Proceedings of the CLEF (Notebook Papers/LABs/Workshops)*, 2010.
- [10] S. Van Dongen, "A cluster algorithm for graphs," Ph.D. dissertation, University of Utrecht, 2000.
- [11] J. Artiles, J. Gonzalo, and S. Sekine, "Weps 2 evaluation campaign: Overview of the web people search clustering task," in *Proceedings of the 2nd Web People Search Evaluation Workshop (WePS 2009)*, 18th World Wide Web Conference, Apr. 2009.