Towards Multi-Level Hybrid Features To Resolve Mixed Entities

Ingyu Lee[1], Byung-Won On[2]
[1]Sorrell College of Business
Troy University, USA
[2]Advanced Institutes of Convergence Technology
Seoul National University, Korea

ABSTRACT: *With the popularity of Internet, tremendous amount of unstructured information becomes available. Consequently, extracting related information from large corpus becomes popular and has been studied by many researchers. However, synonym and polysemy, miss spelling, and using abbreviation make the task difficult. Resolving those confusions is known as an* ***Entity Resolution*** *problem. In this paper, we are proposing a multi-level weighted hybrid feature scheme to resolve mixed entities among unstructured documents. Experimental results show that a weighted hybrid feature improves the accuracy and efficiency.*

Key words: Mixed Entity Resolution, Data Mining, Web Document Clustering, Feature Selections

Received: 2 February 2011, Revised 1 April 2011, Accepted 9 April 2011

1. Introduction

With the advent of Internet, tremendous amount of web pages becomes available to public access. Consequently, extracting related information from large corpus has been studied by many researchers. However, using partial identifier makes it diffi- cult to distinguish different entities which is called an entity *resolution problem*. In addition, spelling errors, synonym and polysemy, and abbreviation make the entity resolution problem much more difficult.

An entity resolution problem is defined as follows: *Given a set of mixed entities* $E=\{e_1,...,e_p,...,e_q,...,e_N\}$ *with the same name description d, group E into K disjoint clusters* $C=\{c_1,...,c_K\}$ *such that entities* $\{e_p^i,...,e_q^i\}$ *within each cluster* c_i *belongs to the same real-world group*. Intuitively, we consider a mixed entity resolution problem as a clustering problem. As the result, clustering algorithms have been popularly used to resolve mixed entities.

Clustering algorithms are categorized into two groups: partition-based and aggregation-based. K-means algorithm is the most popular supervised clustering algorithm [15] based on partition. It initializes centroids according to the given number of clusters and repeatedly computes the distance from centroids. At each iteration, it assigns nodes to the nearest cluster [15]. For example, assume we have N entities, e_1 ,..., e_N and k clusters, C_1 ,..., C_k in the corpus. Then, K-means algorithm repeatedly computes distances from centroids m_1 ,..., m_k and assigns an entity e_i to the nearest cluster C_j . Then, K-means algorithm recomputes centroid m_1 ,..., m_k with new cluster members until algorithm converges (i.e. no membership changes occur). K-means is the most

popular algorithm by its simplicity. However, it requires the number of clusters in advance. On the other hand, hierarchical clustering generates a series of nested clusters by merging simple clusters into larger ones. Assume we have p_1, p_2 ,..., p_N partitions at the first level, then we compute the pairwise distance for each partitions and then two closest partition p_i and p_j are merged into one partition p_{ij} . The algorithm repeatedly merges the closest pairs until it reaches to one partition. Hierarchical clustering is an unsupervised algorithm which does not require the number of clusters in prior. However, it is plausible to be poorly classified since it is not able to reallocate entities [26].

Before applying clustering algorithm, we generate a similarity matrix *A* using features from document corpus. Each column of the matrix indicates a document in the corpus. Term Frequency (TF) and Inverse Document Frequency (IDF) are used to represent each document as a vector. For example, DBLP name data set has co-authors, paper titles, and venues for each document. Then, TF/IDF for each co-authors, paper titles, and venues are used to represent each document. Assume we have *n* textual documents and we want to represent each document d_i with *m* terminologies, then corpus A is represented as a matrix as

$$A_{m\times n} = \begin{pmatrix} | & | & | & | \\ d_1 & d_2 & ... & d_n \\ | & | & | & | \end{pmatrix} \quad (1)$$

where each document d_i is a vector which consists of {$tfidf_{1i}$, $tfidf_{2i}$, $tfidf_{3\,i}$,..., $tfidf_{mi}$}. The component $tfidf_{ji}$ in a vector d_i is a multiplication of tf_{ji} with idf_{ji} for document di. Intuitively, if two documents d_i and d_j share many common terminologies (i.e. highly related), then magnitude of A_{ij} is relatively bigger than others. After constructing similarity matrix *A*, we apply a clustering algorithm to resolve mixed entities. In this paper, we are proposing a multi-level weighted hybrid algorithm to combine different features of documents to construct similarity matrix *A*.

The remainder of this paper is organized as follows. Section 2 describes a framework for mixed entity resolution and details of a multi-level weighted hybrid approach. In Section 3, we describe experimental validation with DBLP name data sets: same spelling but different personnel and the same personnel but different venues. Related works are described in Section 4. Concluding remarks and future plans are followed in Section 5.

2. Methodology

In the previous section, we showed a mixed entity resolution problem is considered as a clustering problem on similarity matrix using TF/IDF (Term Frequency and Inverse Document Frequency). We construct similarity matrices using TF/IDF on co-author lists, paper titles, and venues. Using TF/IDF on coauthor lists shows a better performance than that of using paper titles but a clustering algorithm is not stable with co-author lists. Especially, if a document is written by a single author, then a clustering algorithm could not find a proper cluster. The paper titles TF/IDF matrix shows a stable performance but overall quality is poor compared to using co-author lists TF/IDF. Especially, if authors are working on several different venues (e.g. database, architecture, and network), then it is not easy to distinguish authors. In this paper, we used a multi-level weighted hybrid approach with co-author lists, paper titles, and venues to get benefits from multiple attributes.

In addition, we used two different levels of attribute selection: **Micro-level** and **Macro-level**. Micro-level N-gram method is based on the assumption that parts of spelling error or using abbreviations can be overcome by using an N-gram algorithm rather than using a full terminology. For example, 'John Kim' and 'J. Kim' are treated as the same entity using an N-gram. Using an N-gram generally shows the better accuracy than using a regular TF/IDF with additional cost to compute an N-gram. Macro-level Top-K method is based on the assumption that if two documents are related, they have co-occurrence terminologies or co-occurrence authors. For example, '{*apple, pie, fruit*}' and '{*apple, ipad, company*}' are two different entities. Cooccurring words distinguish the meaning of '{*apple*}.' With a traditional TF/IDF and Micro-Level N-gram, we could not distinguish the semantic difference with a regular TF/IDF. However, Macro-level N-gram could use semantic information as in the given example.

Based on the aforementioned hypothesis, we constructed a similarity matrix with the followings. A term-document matrix *A* is constructed as

$$A_{ij} = TF_{ij} * IDF_{ij} \quad (2)$$

where A_{ij}, TF_{ij} and IDF_{ij} are a term-document matrix value, a term frequency value and an inverse document frequency value for terminology t_i in document d_j, respectively. Then, we created a document-document matrix by multiplying A^T (document-term matrix) with A (term-document matrix). As the results, if two terms are appeared in documents d_i and d_j at the same time, then the multiplication of two values contributes on $A(i, j)$. Otherwise (if two documents do not share a terminology), $A(i, j)$ is set to zero as in

$$A(i,j) = \sum_{t \in d_i \cap d_j} d(t,i) \times d(t,j) \tag{3}$$

Intuitively, document d_i is strongly related with document d_j when two documents are sharing many terminologies together. Otherwise, the similarity value $A(i, j)$ becomes zero.

In a multi-level weighted hybrid scheme, we separately generate TD/IDF matrix for author names and paper titles. Then, we combine two different levels of author and title matrices with different weight values: N-gram and Top-K. We tried three different types of N-grams: 3-grams, 4-grams, and 5-grams. We also constructed Top-K co-occurrence matrices based on the assumption that two different terminologies are used in different documents, then two documents are strongly related and the cooccurrence terminologies can be used as a feature to distinguish semantic. To generate Top-K matrices, we sorted the terminology by decreasing order of frequency, and then carefully selecting terminologies one by one. Finally, we computed the pair wise TF/IDF values by multiplying two TF/IDF values for Top- K terminologies.

In our scheme, the similarity matrix A is defined as

$$A = \sum_{i=1,2} (u * TD_i + v * GD_i) \tag{4}$$

where u and v are weighting factors, TD_i is a Top-K macrolevel TF/IDF, and GD_i is an N-gram TF/IDF. To get an optimal weighting factor, we construct a matrix framework such as

$$L = || A - X W X^T ||_F^2 + \lambda || W - I ||_F^2 \tag{5}$$

where A is a term document matrix and W is a diagonal matrix whose values are weight w_i. Matrix X is a corresponding cluster matrix based on training data. Matrix W is a diagonal matrix with a weighting vector $w = (u, v)$ of two different matrices. To get an optimal weight value (for the given training set), we take partial derivatives for (u, v) and set to zero. Solution of equations

$$\frac{\partial L}{\partial u} = \frac{\partial L}{\partial v} = 0 \tag{6}$$

is an optimal weighting factors for the given training set. The equation is represented as a system of linear equation in

$$(X X^T X X^T + \lambda I) W = A X X^T + \lambda I \tag{7}$$

3. Experimental Validation

To measure the performance of different features, we used *DBLP* author name data set as shown in Table 1. Name set data has the same spelling 'Wei Wang' but each author is a different personnel. In addition, the cluster size is extremely skewed. For example, one cluster has only one member document but another cluster includes 91 documents. Therefore, distinguishing each author from the given name data set is much more difficult. To evaluate the proposed method, we measured precision, recall, and F-measure using the author name data set.

Precision is defined as the number of entities correctly clustered divided by the number of entities in the cluster as in

$$P = \frac{TruePositive}{(TruePositive + FalseP\ ositive)} \tag{8}$$

Recall is defined as the number of correctly clustered entities divided by the number of entities in the solution set as in

$$R = \frac{TruePositive}{(TruePositive + FalseP\ ositive)} \quad (9)$$

Precision and recall are known as biased based on the size of clusters. Precision is relatively high when cluster size is small and recall shows the opposite tendency. To balance the latter, we also used an F-measure which is an arithmetic mean of precision and recall as in

$$F = \frac{2PR}{(P + R)} \quad (10)$$

ID	Docs	Description
Wei Wang 1	1	Fudan University
Wei Wang 2	2	MIT
Wei Wang 3	5	U. of Maryland
Wei Wang 4	2	U. of Naval Engineering
Wei Wang 5	1	Chinese Academy
Wei Wang 6	2	Rutgers University
Wei Wang 7	11	Purdue University
Wei Wang 8	16	INRIA
Wei Wang 9	4	Peking University
Wei Wang 10	3	NU of Singapore
Wei Wang 11	3	Nanyang Tech.
Wei Wang 12	20	U. of Nebraska
Wei Wang 13	36	U. of New South Wales
Wei Wang 14	4	Language Weaver, Inc.
Wei Wang 15	3	Chinese U. of Hong Kong
Wei Wang 16	2	Zhejiang University
Wei Wang 17	66	Fudan Unverrsity
Wei Wang 18	91	U. of North Carolina
Total	272	

Table 1. Author Name Data Set I

Table 2 shows the performance results with name data set using only **Title** field as a feature vector. Among different methods, using an N-gram feature shows a slightly better precision than those of using other features. However, the recall for an N-gram feature is worse than those of using others. Especially, 5-gram shows the best performance in terms of precision with the worst recall. Combining two or three features with the same weight worsen the performance. However, to get benefit from N-gram with similar recall, we used different weights for each feature. As we expected, the results shows a better precision with a similar recall. Recall is still a relatively lower compared to that of 5-gram. We conjecture that high number of clusters in corpus (i.e. 19 in our experimental data) worsen the the recall.

Table 3 shows the performance results with **Authors** property feature. As we expected, using co-author lists shows much better performance than those of using paper titles. Since the same group of authors prefers to work together repeatedly, coauthor lists is a good entity resolution feature than paper titles. Among six different methods, N-gram shows the highest precision without losing performance in recall. We assume that author names are relatively shorter in length than paper titles which fits better with N-gram algorithm. Combining different features together with different weight values shows the similar performance for our name data set. We conjecture that coauthor lists itself shows high performance and could not improve performance by adding other features. However, we still have a difficulty to distinguish a single authored paper. We conjecture that the latter can be corrected by adding a paper title field as a feature.

Table 4 shows the experimental results of using weighted hybrid of Titles and Authors property. Since using co-author lists

Features	Precision	Recall	Fmeasure
Normal TF/IDF	0.61	0.27	0.37
Micro (3-gram) TF/IDF	0.63	0.22	0.32
Micro (4-gram) TF/IDF	0.63	0.22	0.32
Micro (5-gram) TF/IDF	0.67	0.19	0.30
Macro (top-10) TF/IDF	0.62	0.27	0.37
Macro (top-20) TF/IDF	0.62	0.26	0.36
Macro (top-30) TF/IDF	0.62	0.24	0.35
Normal + Micro (5-gram) TF/IDF	0.62	0.25	0.36
Normal + Macro (top-30) TF/IDF	0.62	0.24	0.35
Normal + Micro (5-gram) + Macro (top-30) TF/IDF	0.65	0.25	0.37

Table 2. Experimental Results for Author Name Data Set using **Title** property

Features	Precision	Recall	Fmeasure
Normal TF/IDF	0.88	0.40	0.56
Micro (3-gram) TF/IDF	0.91	0.37	0.53
Micro (4-gram) TF/IDF	0.92	0.36	0.52
Micro (5-gram) TF/IDF	0.93	0.43	0.59
Macro (top-10) TF/IDF	0.89	0.40	0.54
Macro (top-20) TF/IDF	0.89	0.38	0.53
Macro (top-30) TF/IDF	0.90	0.43	0.58
Normal + Micro (5-gram) TF/IDF	0.88	0.43	0.58
Normal + Macro (top-30) TF/IDF	0.89	0.43	0.58
Normal + Micro (5-gram) + Macro (top-30) TF/IDF	0.92	0.32	0.57

Table 3. Experimental Results for Author Name Data Set using **Authors** property

Features	Precision	Recall	Fmeasure
Normal TF/IDF	0.89	0.41	0.56
Micro (3-gram) TF/IDF	0.93	0.41	0.57
Micro (4-gram) TF/IDF	0.92	0.38	0.54
Micro (5-gram) TF/IDF	0.94	0.35	0.51
Macro (top-10) TF/IDF	0.88	0.39	0.55
Macro (top-20) TF/IDF	0.90	0.40	0.56
Macro (top-30) TF/IDF	0.90	0.39	0.54
Normal + Micro (5-gram) TF/IDF	0.90	0.41	0.57
Normal + Macro (top-30) TF/IDF	0.89	0.39	0.55
Normal + Micro (5-gram) + Macro (top-30) TF/IDF	0.90	0.40	0.55

Table 4. Experimental Results for Author Name Data Set using **Author** and **Title** properties

shows better performance than those of using paper titles, we used 0.8 for co-author lists and 0.2 for paper titles which are the close number we computed based on the optimization matrix framework described in the previous section. Since an optimal

Author	Category	Documents(C1/C2)
Dongwon Lee	2	30(25/5)
Wei Cai	2	7 (5/2)
H Cai	2	5 (4/1)
Jian Li	2	21(18/3)
Yuan Xie	2	20(19/1)
Jia Li	2	27(24/3)
Peng Liu	2	32(25/7)
Hui Song	2	6 (5/1)
Lin Li	2	11(8/3)
Murali Mani	2	11(9/2)
James Ze Wang	2	33(24/9)
Sanghyun Park	2	18(16/2)
Li Chen	2	60(38/22)
Prasenjit Mitra	2	11(10/1)
Zhenyu Liu	2	8 (4/4)
John M. Carroll	2	92(86/6)

Table 5. Author Name Data Set II

	Hybrid Feature			Author Only		
Author	**Precision**	**Recall**	**Fmeasure**	**Precision**	**Recall**	**Fmeasure**
Dongwon Lee	0.91	0.50	0.64	0.85	0.50	0.63
Wei Cai	0.80	0.50	0.61	0.75	0.50	0.60
H Cai	0.75	0.50	0.60	0.87	0.50	0.63
Jian Li	0.91	1.00	0.91	0.90	1.00	0.01
Yuan Xie	0.97	0.50	0.66	0.94	0.50	0.65
Jia Li	0.93	0.50	0.65	0.89	0.50	0.64
Peng Liu	0.87	0.50	0.63	0.83	0.50	0.62
Hui Song	0.90	0.50	0.64	1.00	0.60	0.75
Lin Li	0.90	0.56	0.69	1.00	0.68	0.81
Murali Mani	0.90	0.50	0.64	1.00	0.61	0.75
James Ze Wang	0.91	0.58	0.71	0.80	0.50	0.61
Sanghyun Park	0.97	1.00	0.97	0.93	1.00	0.93
Li Chen	0.91	1.00	0.91	0.78	0.86	0.82
Prasenjit Mitra	0.94	0.50	0.65	0.92	0.50	0.65
Zhenyu Liu	0.78	0.62	0.69	0.78	0.62	0.69
John M. Carroll	0.96	0.50	0.65	0.92	0.50	0.64

Table 6. Experimental Results for the Name Data Set II

weight can be changed based on the training data set, we used only a small set as a training rather than the whole data set. The performance results is similar to or slightly better than that of using co-author lists alone. In addition, 5-gram method shows the

best precision with a decent recall. Since *DBLP* name data set is already cleaned and fixed errors such as typos and misspelling, the benefits of using a hybrid features is marginal at best. However, for a single authored paper, a hybrid method shows better performance to distinguish than other methods. We conjecture that a hybrid method will show better performance when corpus has a little bit errors such as typos and spelling errors which are common in real data set.

The second name data set has 16 different authors who have worked in two different venues. The data set is extremely skewed. Even all authors have two different venues in research, the number of papers in one venue dominates the cluster as shown in the table. The goal is to distinguish the venue using co-author lists, paper titles, and journal titles. Since each document belongs to the same author even it belongs to two different clusters, resolving records is much more difficult. Sometimes, the authors may have the same set of co-author lists even the venue is different. We conjecture that hybrid features can overcome by considering different perspective for a document.

Table 6 shows the experimental results of the second name data set. Since each document belongs to the same author, using co-author list only has limitations to distinguish each document. In general, combining co-author list with title shows a better performance than using only co-author list. For some data set, using co-author list only shows better performance than hybrid approach. We conjecture that author with different co-author list on two different venues got benefits with co-author list.

4. Related Works

Many researches have been done to resolve mixed entities. Bekkerman et al. in [4] proposed methods to disambiguate namesakes that appear in the web using link structure of web pages. The authors uses a multi-way distributional clustering method. Monkov et. al. used a lazy graph walk algorithm to disambiguate namesakes in email documents in their paper [5]. Banerjee et. al. proposed a multi-way clustering method in relation graphs in [3]. Different types of entities are simultaneously clustered based not only on their intrinsic attribute values but also on the multiple relations between entities. Han et al. in [22] proposed supervised learning-based approaches including Naive Bayes Model and using Support Vector Machine. The authors also proposed a K-way spectral clustering method to resolve mixed entities. Since spectral clustering considers global connectivity, the proposed method shows better performance for overlapped venue or authors. Malin [25] utilized hierarchical clustering methods on the exact name similarity.

In real name data set, the corpus becomes larger as in Internet. To handle a large number of name entities, scalable algorithms are needed. Lee et. al. in [24] proposed a scalable citation labeling algorithm based on sampling-based technique to quickly determine a small number of candidates from the entire author names in a digital library. On et. al. [29] proposed a multi-level methods to resolve mixed entities. In their paper, authors proposed using a multi-level graph partitioning algorithm which scales with O(logN) complexity.

To the best of our knowledge, our paper is the first approach to build a weighted hybrid scheme of multi-level features to resolve mixed entities. Using only one attribute feature may be limited by typos, miss spellings, polysemy and synonym, and single authored paper. However, combining several attributes with different weights can avoid the aforementioned problems. Especially, two different levels (Micro and Macro) gives benefits to resolve mixed entities.

5. Conclusion

To resolve a mixed entity problem, we proposed a multi-level weighted hybrid scheme. Using co-author list TF/IDF performs better than using paper title TF/IDF in our experiments. It also improves reliability considering name data set is extremely skewed. In addition, we provided a macro level Top-K scheme and micro level N-gram scheme. The micro level N-gram shows the better performance when the terminology is in short length and the corpus having spelling error, and abbreviations. The macro level Top-K scheme can detect the semantical difference by using co-occurrent terminologies. The experimental results shows that the proposed multi-level hybrid method keeps the precision with a similar recall.

The current version of multi-level weighted hybrid approach is based on a supervised algorithm. In reality, an unsupervised method is more suitable in a mixed entity problem. In the near future, we will develop a semi-supervised algorithm based on the feedback from user experiences. Estimating the number of clusters is also another challenging problem in cluster analysis. We are planning to provide an algorithm to estimate the number of clusters based on the connectivity of the input data. At last,

precision and recall are not the suitable to measure the performance of mixed entity resolving algorithm, we are planning to provide a better quality metric to measure the performance.

References

[1] Bekkerman, R. (2005). Name Data Set, http://www.cs.umass.edu/~ronb

[2] Elmacioglu, E., Tan, Y., Yan, S., Kan, M., Lee, D. (2007). PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features, Proceedings of InternationalWorkshop on Semantic Evaluation (SemEval), Prague, Czech Republic.

[3] Banerjee, A., Basu, S., Merugu, S. (2007). Multi-way clustering on relation graphs, Proceedings of SIAM Data Mining.

[4] Bekkerman, R., McCallum, A. (2005). Disambiguating web appearances of people in a social network, *In*: Proceedings of International Conference on World Wide Web.

[5] Monkov, E., Cohen, W., Ng, A. (2006). Contextual search and name disambiguation in email using graphs, Proceedings of SIGIR.

[6] Han, J., Kamber, M., Tung, A. (2001). Spatial clustering methods in data mining: a survey, Geographic Data Mining and Knowledge Discovery, Taylor and Francis.

[7] MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations, *In*: Proceedings of the Fifth Berkeley Sympositum on Mathematical Statistics and Probability.

[8] Pothen, A., Simon, H., Liou, K. (1990). Partitioning sparse sparse matrices with eigenvectors of graphs, *SIAM Journal on Matrix Analysis and Applications*, 11 (3) 430 – 452.

[9] Karypis, G., Kumar, V. (1998). A parallel algorithm for multilevel graph partitioning and sparse matrix ordering, *Journal of Parallel and Distributed Computing*, 48 (1) 71 – 95.

[10] Karypis, G., Kumar, V. (1997). ParMETIS: Parallel graph partitioning and sparse matrix ordering library, Department of Computer Science, University of Minnesota, TR 97-60.

[11] Fiedler, M. (1973). Algebraic connectivity of graphs, *Czechoslovak Math Journal*, 23: 298 – 305.

[12] Dunlup, A., Kernighan, B. (1985). A procedure for placement of standard-cell VLSI circuits, *IEEE Tans. CAD*, 92 – 98.

[13] Fiduccia, C., Mattheyses, R. (1982). A linear time heuristic for improving network partitions, *In*: Proceedings of 19th IEEE Design Automation Conference.

[14] Heath, M. (2002). Scientific computing: an introductory survey, Prentice Hall.

[15] Han, J., Kamber, M., Tung, A. (2001). Spatial clustering methods in data mining: A survey. Taylor and Francis.

[16] Golub, G.., Loan, C. (1996). Matrix Computations, Johns Hopkins University Press, Baltimore, MD, US, 3rd edition.

[17] Zeimpekis, D., Gallopoulos, E. (2006). TMG: A MATLAB toolbox for generating term document matrices from text collections, Grouping Multidimensional Data: Recent Advances in Clustering, Springer 187 – 210.

[18] Chua, F. (2009). Dimensionality Reduction and Clustering of Text Documents. Technical Report, Singapore Management University.

[19] Hendrickson, B., Leland, R. (1994). The Chaco user's guide: version 2.0, Sandia.

[20] Dhillon, I., Guan, Y., Kulis, B. (2007). Weighted graph cuts without eigenvectors: A multilevel approach, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29 (11) 1944 – 1957.

[21] Cheng, D., Kannan, R., Vempala, S., Wang, G. (2005). A divideand- merge methodology for clustering, *ACM Transactions on Database Systems*.

[22] Han, H., Giles, C., Zha, H. (2004). Two supervised learning approaches for name disambiguation in author citations, *In*: Proceedings of ACM/IEEE Joint Conference on Digital Libraries.

[23] Han, H., Giles, C., Zha, H. (2005). Name disambiguation in author citations using a k-way spectral clustering method, *In*: Proceedings of ACM/IEEE Joint Conference on Digital Libraries.

[24] Lee, D., On, B., Kang, J., Park, S. (2005). Effective and scalable solutions for mixed and split citation problems in digital libraries, *In*: Proceedings of the ACM SIGMOD Workshop on Information Quality in Information Systems, Baltimore, MD, USA.

[25] Malin, B. (2005). Unsupervised name disambiguation via social network similarity, Proceedings of the SIAM SDM Workshop on Link Analysis, Counterterrorism and Security.

[26] Jain, A. (2008). Data clustering: 50 years beyond K-means, Proceedings of the 19th International Conference on Pattern Recognition (ICPR), Tampa, FL, USA.

[27] Cohen, W., Ravikumar, P., Fienberg, S. (2003). A comparison of string distance metrics for name-matching tasks, *In*: Proceedings of the IIWEB workshop.

[28] Newman, M. (2004). Detecting community structure in networks, *European Physics Journal B*(38) 321–330

[29] On, B., Lee, I., Lee, D. (2011). Scalable Clustering Methods for the Name Disambiguation Problem, Knowledge and Information Systems. 6